CHAPTER

Quantum case statements

6

Quantum programs in the superposition-of-data paradigm have been systematically investigated in Chapters 3 to 5. In particular, in Chapter 3, we studied quantum **while**-programs and recursive quantum programs, and showed how some quantum algorithms can be conveniently written as this kind of quantum programs. The control flow of a quantum **while**-program is generated by the case statements and **while**-loops within it, and the control flow of a recursive quantum program is further produced by procedure calls. Since the information that determines the control flow of a quantum **while**-program or a recursive quantum program is classical rather than quantum, such a control flow was properly named a classical control flow (in a quantum program).

The aim of this chapter and the next is to introduce quantum programs in the superposition-of-programs paradigm: in other words, quantum programs with quantum control flows. As we know, the control flow of a program is determined by those program constructs within it such as case statement, loop and recursion. Interestingly, the notions of case statement, loop and recursion in classical programming split into two different versions in the quantum setting:

(i) case statement, loop and recursion with classical control, which have been discussed in detail in Chapters 3 to 5;
(ii) quantum case statement, loop and recursion – case statement, loop and recursion with *quantum control*.

As we will see later, a large class of quantum algorithms can be much more conveniently programmed in a programming language with quantum control. This chapter focuses on quantum case statements. Loop and recursion with quantum control flow will be discussed in the next chapter.

This chapter is organized as follows:

- In Section 6.1, the notion of quantum case statement is carefully motivated through the example of quantum walk on a graph. The control flow of a quantum case statement is then analyzed, and the technical difficulty in defining the semantics of quantum case statement is unveiled.

- A new quantum programming language QuGCL is defined in Section 6.2 to support programming with quantum case statements.
- Section 6.3 prepares several key ingredients needed in defining the denotational semantics of QuGCL, including guarded composition of various quantum operations. The denotational semantics of QuGCL is presented in Section 6.4.
- In Section 6.5, the notion of quantum choice is defined based on quantum case statement.
- A family of algebraic laws for QuGCL programs are presented in Section 6.6. They can be used in verification, transformation and compilation of programs with quantum case statements.
- A series of examples are presented in Section 6.7 to illustrate the expressive power of the language QuGCL.
- The possible variants and generalizations of quantum case statements are discussed in Section 6.8.
- The proofs of some lemmas, propositions and theorems in Sections 6.3 to 6.6 are tedious. For readability, they are deferred to the last section, Section 6.9.

6.1 CASE STATEMENTS: FROM CLASSICAL TO QUANTUM

Let us start with an intuitive discussion about the following questions: Why should we introduce the new notion of quantum case statement? How does it differ from the case statement in the quantum programs considered in Chapter 3? We answer these questions and thus motivate the notion of quantum case statement in three steps:

(i) ***Case statement in classical programming:*** Recall that a conditional statement in classical programming is written as

$$\mathbf{if}\ b\ \mathbf{then}\ S_1\ \mathbf{else}\ S_0\ \mathbf{fi} \tag{6.1}$$

where b is a Boolean expression. When b is true, subprogram S_1 will be executed; otherwise, S_0 will be executed. More generally, a case statement in classical programming is a collection of guarded commands written as

$$\mathbf{if}\ (\square\, i \cdot G_i \rightarrow S_i)\ \mathbf{fi} \tag{6.2}$$

where for each $1 \leq i \leq n$, the subprogram S_i is guarded by the Boolean expression G_i, and S_i will be executed only when G_i is true.

(ii) ***Classical case statement in quantum programming:*** A notion of a classical case statement in quantum programming was defined in Chapter 3 based on a quantum measurement. Let $\overline{q}$ be a family of quantum variables and $M = \{M_m\}$ a measurement on $\overline{q}$. For each possible measurement outcome m, let S_m be a quantum program. Then a case statement can be written as follows:

$$\mathbf{if}\ (\square\, m \cdot M[\overline{q}\,] = m \rightarrow S_m)\ \mathbf{fi} \tag{6.3}$$

The statement (6.3) selects a command according to the outcome of measurement M: if the outcome is m, then the corresponding command S_m will be executed. In particular, whenever M is a yes-no measurement, that is, it has only two possible outcomes 1 (yes) and 0 (no), then case statement (6.3) is a generalization of conditional statement (6.1), and it can be appropriately termed as a classical conditional statement in quantum programming. As indicated by Exercise 3.4.1, quantum **while**-loop (3.4) can be seen as a recursive program declared by such a conditional statement.

(iii) ***Quantum case statement***: In addition to (6.3), there is actually another kind of case statement, which is very useful in quantum programming. This new notion of case statement can be defined by extending a key idea from the definition of the shift operator of a quantum walk on a graph. Recall from Example 2.3.2 that the shift operator is an operator in $\mathcal{H}_d \otimes \mathcal{H}_p$ defined as follows:

$$S|i, v\rangle = |i, v_i\rangle$$

for each direction $1 \leq i \leq n$ and each vertex $v \in V$, where v_i is the ith neighbor of v, and $\mathcal{H}_d$, $\mathcal{H}_p$ are the direction "coin" space and the position space, respectively.

This shift operator can be viewed in a slightly different way: for each $1 \leq i \leq n$, we define the shift S_i in the direction i as an operator in $\mathcal{H}_p$:

$$S_i|v\rangle = |v_i\rangle$$

for any $v \in V$. Then we are able to combine these operators S_i $(1 \leq i \leq n)$ along the "coin" to form the whole shift operator S:

$$S|i, v\rangle = |i\rangle S_i|v\rangle \tag{6.4}$$

for any $1 \leq i \leq n$ and $v \in V$. It is worth noticing that operators S and S_i $(1 \leq i \leq n)$ are defined in different Hilbert spaces: S is in $\mathcal{H}_d \otimes \mathcal{H}_p$, whereas all S_i are in $\mathcal{H}_p$.

Let us carefully observe the behavior of shift operator S. The operators $S_1, \ldots, S_n$ can be seen as a collection of programs independent of each other. Then S can be seen as a kind of case statement of $S_1, \ldots, S_n$ because S selects one of them for execution. But equation (6.4) clearly indicates that this case statement is different from (6.3): the selection in equation (6.4) is made according to the basis state $|i\rangle$ of the "coin space," which is quantum information rather than classical information. Thus, we can appropriately call S a *quantum case statement*. At this stage, the reader might still not be convinced that the behaviors of case statement (6.3) and a quantum case statement are really different, despite the fact that measurement outcomes m are classical information, and basis states $|i\rangle$ are quantum information. The essential difference between them will become clearer later, when we consider the control flow of a quantum case statement.

The preceding idea can be significantly extended. Let $S_1, S_2, \ldots, S_n$ be a collection of general quantum programs whose state spaces are the same Hilbert space $\mathcal{H}$. We introduce an external quantum system, called the "coin" system. It is allowed to be a single system or a composite system, so denoted by a quantum register $\overline{q}$ consisting of a family of new quantum variables that do not appear in $S_1, S_2, \ldots, S_n$. Assume that the state space of system $\overline{q}$ is an n-dimensional Hilbert space $\mathcal{H}_{\overline{q}}$ and $\{|i\rangle\}_{i=1}^n$ is an orthonormal basis of it. Then a quantum case statement S can be defined by combining programs $S_1, S_2, \ldots, S_n$ along the basis $\{|i\rangle\}$:

$$\begin{array}{rll} S \equiv \mathbf{qif}\,[\overline{q}]: & |1\rangle \rightarrow S_1 & \\ \square & |2\rangle \rightarrow S_2 & \\ & \ldots\ldots & \qquad (6.5) \\ \square & |n\rangle \rightarrow S_n & \\ \mathbf{fiq} & & \end{array}$$

or more compactly

$$S \equiv \ \mathbf{qif}\,[\overline{q}](\square i \cdot |i\rangle \rightarrow S_i)\ \mathbf{fiq}$$

Quantum Control Flows:

Now let us look at the control flow of quantum case statement (6.5) – the order of its execution. The control flow of S can be clearly seen through its semantics. Following equation (6.4), it is reasonable to conceive that the semantics $[\![S]\!]$ of S should be defined in the tensor product $\mathcal{H}_{\overline{q}} \otimes \mathcal{H}$ by

$$[\![S]\!](|i\rangle|\varphi\rangle) = |i\rangle([\![S_i]\!]|\varphi\rangle) \tag{6.6}$$

for every $1 \le i \le n$ and $|\varphi\rangle \in \mathcal{H}$, where $[\![S_i]\!]$ is the semantics of S_i. Then the control flow of program S is determined by "coin" variables $\overline{q}$. For each $1 \le i \le n$, S_i is guarded by the basis state $|i\rangle$. In other words, the execution of program (6.5) is controlled by "coin" $\overline{q}$: when $\overline{q}$ is in state $|i\rangle$, subprogram S_i will be executed. The really interesting thing here is that $\overline{q}$ is a quantum "coin" rather than a classical "coin," and thus it can be not only in the basis states $|i\rangle$ but also in a superposition of them. A superposition of these basis states yields a quantum control flow – *superposition of control flows*:

$$[\![S]\!]\left(\sum_{i=1}^{n} \alpha_i |i\rangle|\varphi_i\rangle\right) = \sum_{i=1}^{n} \alpha_i |i\rangle([\![S_i]\!]|\varphi_i\rangle) \tag{6.7}$$

for all $|\varphi_i\rangle \in \mathcal{H}$ and complex numbers α_i $(1 \le i \le n)$. Intuitively, in equation (6.7), for every $1 \le i \le n$, subprogram S_i is executed with probability amplitude α_i. This is very different from the classical case statement (6.3) of quantum programs where different guards " $M[\overline{q}] = m_1$", ..., "$M[\overline{q}] = m_n$" cannot be superposed.

Technical Difficulty in Defining Semantics of Quantum Case Statements: At first glance, it seems that the defining equation of shift operator in a quantum walk can be smoothly generalized to equation (6.6) to define the denotational semantics of a general quantum case statement. But there is actually a major (and subtle) difficulty in equation (6.6). For the case where no quantum measurements occur in any S_i $(1 \leq i \leq n)$, the operational semantics of each S_i is simply given as a sequence of unitary operators, and equation (6.6) is not problematic at all. Whenever some S_i contains quantum measurements, however, its semantic structure becomes a tree of linear operators with branching happening at the points where the measurements are performed. Then equation (6.6) becomes meaningless within the framework of quantum mechanics, and defining the semantics of quantum case statement S requires properly combining a collection of trees of quantum operations such that the relevant quantum mechanical principles are still obeyed. This problem will be circumvented in Sections 6.3 and 6.4 by introducing a semi-classical semantics in terms of operator-valued functions.

Exercise 6.1.1. *Why can equation (6.6) not properly define the semantics of quantum case statement (6.5) when a quantum measurement appears in some of S_i $(1 \leq i \leq n)$? Give some example(s) to illustrate your argument.*

6.2 QuGCL: A LANGUAGE WITH QUANTUM CASE STATEMENT

The notion of a quantum case statement was carefully motivated in the previous section. Now we start to study programming with the quantum case statement. First of all, we formally define a programming language QuGCL with the program construct of quantum case statement. It can be seen as a quantum counterpart of Dijkstra's GCL (Guarded Command Language). The alphabet of QuGCL is given as follows:

- As in Chapter 3, we assume a countable set $qVar$ of quantum variables ranged over by $q, q_1, q_2, \ldots$. For each quantum variable $q \in qVar$, its type is a Hilbert space $\mathcal{H}_q$, which is the state space of the quantum system denoted by q. For a quantum register $\overline{q} = q_1, q_2, \ldots, q_n$ of distinct quantum variables, we write:

$$\mathcal{H}_{\overline{q}} = \bigotimes_{i=1}^{n} \mathcal{H}_{q_i}.$$

- For simplicity of the presentation, QuGCL is designed as a purely quantum programming language, but we include a countably infinite set Var of classical variables ranged over by $x, y, \ldots$ so that we can use them to record the outcomes of quantum measurements. However, classical computation described by, for example, the assignment statement $x := e$ in a classical programming language, is excluded. For each classical variable $x \in Var$, its type is assumed to be a non-empty set D_x; that is, x takes values from D_x. In applications, if x is used to store the outcome of a quantum measurement M, then all possible outcomes of M should be in D_x.

- The sets of classical and quantum variables are required to disjoint: $qVar \cap Var = \emptyset$.

Using the alphabet presented here, we can define programs in QuGCL. For each QuGCL program S, we write $var(S)$ for the set of its classical variables, $qvar(P)$ for its quantum variables and $cvar(P)$ for its "coin" variables.

Definition 6.2.1. *QuGCL programs are inductively defined as follows:*

(i) **abort** *and* **skip** *are programs, and*

$$\begin{aligned} var(\mathbf{abort}) &= var(\mathbf{skip}) = \emptyset, \\ qvar(\mathbf{abort}) &= qvar(\mathbf{skip}) = \emptyset, \\ cvar(\mathbf{abort}) &= cvar(\mathbf{skip}) = \emptyset. \end{aligned}$$

(ii) *If $\overline{q}$ is a quantum register, and U is a unitary operator in $\mathcal{H}_{\overline{q}}$, then*

$$\overline{q} := U[\overline{q}]$$

is a program, and

$$var(\overline{q} := U[\overline{q}]) = \emptyset, \quad qvar(\overline{q} := U[\overline{q}]) = \overline{q}, \quad cvar(\overline{q} := U[\overline{q}]) = \emptyset.$$

(iii) *If S_1 and S_2 are programs such that $var(S_1) \cap var(S_2) = \emptyset$, then $S_1; S_2$ is a program, and*

$$\begin{aligned} var(S_1; S_2) &= var(S_1) \cup var(S_2), \\ qvar(S_1; S_2) &= qvar(S_1) \cup qvar(S_2), \\ cvar(S_1; S_2) &= cvar(S_1) \cup cvar(S_2). \end{aligned}$$

(iv) *If $\overline{q}$ is a quantum register, x is a classical variable, $M = \{M_m\}$ is a quantum measurement in $\mathcal{H}_{\overline{q}}$ such that all possible outcomes of M are in D_x, and $\{S_m\}$ is a family of programs indexed by the outcomes m of measurement M such that $x \notin \bigcup_m var(S_m)$, then the classical case statement of S_m's guarded by measurement outcomes m's:*

$$S \equiv \mathbf{if}\ (\square m \cdot M[\overline{q} : x] = m \rightarrow S_m)\ \mathbf{fi} \tag{6.8}$$

is a program, and

$$\begin{aligned} var(S) &= \{x\} \cup \left(\bigcup_m var(S_m)\right), \\ qvar(S) &= \overline{q} \cup \left(\bigcup_m qvar(S_m)\right), \\ cvar(S) &= \bigcup_m cvar(S_m). \end{aligned}$$

(v) *If $\overline{q}$ is a quantum register, $\{|i\rangle\}$ is an orthonormal basis of $\mathcal{H}_{\overline{q}}$, and $\{S_i\}$ is a family of programs indexed by the basis states $|i\rangle$'s such that*

$$\overline{q} \cap \left(\bigcup_i qvar(S_i) \right) = \emptyset,$$

then the quantum case statement of S_i's guarded by basis states $|i\rangle$'s:

$$S \equiv \textbf{qif}\ [\overline{q}]\ (\square i \cdot\ |i\rangle \to S_i)\ \textbf{fiq} \tag{6.9}$$

is a program, and

$$var(S) = \bigcup_i var(S_i),$$

$$qvar(S) = \overline{q} \cup \left(\bigcup_i qvar(S_i) \right),$$

$$cvar(S) = \overline{q} \cup \left(\bigcup_i cvar(S_i) \right).$$

This definition looks quite complicated with quantum programs and their classical, quantum and "coin" variables being defined simultaneously. But the syntax of QuGCL can be simply summarized as follows:

$$\begin{array}{lll} S := & \textbf{abort} \mid \textbf{skip} \mid \overline{q} := U[\overline{q}] \mid S_1; S_2 & \\ & \mid \textbf{if}\ (\square m \cdot M[\overline{q} : x] = m \to S_m)\ \textbf{fi} & \text{(classical case statement)} \\ & \mid \textbf{qif}\ [\overline{q}](\square i \cdot |i\rangle \to S_i)\ \textbf{fiq} & \text{(quantum case statement)} \end{array}$$

For simplicity, we often write $U[\overline{q}]$ for the unitary statement $\overline{q} := U[\overline{q}]$. The meanings of **skip**, unitary transformation and sequential composition in QuGCL are the same as in the quantum **while**-language defined in Chapter 3. As in Dijkstra's GCL, **abort** is the undefined instruction that can do anything and even does not need to terminate. The intuitive meaning of quantum case statement (6.9) was already carefully explained in Section 6.1. But several delicate points in the design of the language QuGCL deserve careful explanations:

- The requirement $var(S_1) \cap var(S_2) = \emptyset$ in the sequential composition $S_1; S_2$ means that the outcomes of measurements performed at different points are stored in different classical variables. Such a requirement is mainly for technical convenience, and it will considerably simplify the presentation.
- The statements (6.8) and (6.3) are essentially the same, and the only difference between them is that a classical variable x is added in (6.8) to record the measurement outcome. It is required in statement (6.8) that $x \notin \bigcup_m var(S_m)$. This means that the classical variables already used to record the outcomes of the measurements in S_m's are not allowed to store the outcome of a new measurement. This technical requirement is cumbersome, but it can significantly

simplify the presentation of the semantics of QuGCL. On the other hand, it is not required that the measured quantum variables $\overline{q}$ do not occur in S_m. So, measurement M can be performed not only on an external system but also on some quantum variables within S_m.

- It should be emphasized that in the quantum case statement (6.9) the variables in $\overline{q}$ are not allowed to appear in any S_i's. This indicates that the "coin system" $\overline{q}$ is *external* to programs S_i's. This requirement is so important that it will be emphasized again and again in this chapter and the next. The reason for this requirement will become clear when we consider the semantics of quantum case statement in the following two sections.
- Obviously, all "coins" are quantum variables: $cvar(S) \subseteq qvar(S)$ for all programs S. It will be needed in defining a kind of equivalence between quantum programs to distinguish the set $cvar(S)$ of "coin" variables from other quantum variables in S.

6.3 GUARDED COMPOSITIONS OF QUANTUM OPERATIONS

The syntax of quantum programming language QuGCL was defined in the last section. Now we consider how to define the semantics of QuGCL. It is clear that the main issue in defining the semantics of QuGCL is the treatment of the quantum case statement, because the semantics of the other program constructs either are trivial or were already well-defined in Chapter 3. As was pointed out in Section 6.1, a major difficulty in defining the semantics of a quantum case statement emerges in the case where quantum measurements occur in some of its branch subprograms. So, in this section, we prepare the key mathematical tool – guarded composition of quantum operations – for overcoming this difficulty.

6.3.1 GUARDED COMPOSITION OF UNITARY OPERATORS

To ease the understanding of a general definition of guarded composition, we start with a special case of the guarded composition of unitary operators, which is a straightforward generalization of the quantum walk shift operator S in equation (6.4). In this easy case, no quantum measurements are involved.

Definition 6.3.1. *For each $1 \leq i \leq n$, let U_i be a unitary operator in Hilbert space $\mathcal{H}$. Let $\mathcal{H}_q$ be an auxiliary Hilbert space, called the "coin space," with $\{|i\rangle\}$ as an orthonormal basis. Then we define a linear operator U in $\mathcal{H}_q \otimes \mathcal{H}$ by*

$$U(|i\rangle|\psi\rangle) = |i\rangle U_i|\psi\rangle \tag{6.10}$$

for any $|\psi\rangle \in \mathcal{H}$ and for any $1 \leq i \leq n$. By linearity we have:

$$U\left(\sum_i \alpha_i|i\rangle|\psi_i\rangle\right) = \sum_i \alpha_i|i\rangle U_i|\psi_i\rangle \tag{6.11}$$

for any $|\psi_i\rangle \in \mathcal{H}$ and complex numbers α_i. The operator U is called the guarded composition of U_i $(1 \le i \le n)$ along the basis $\{|i\rangle\}$ and written as

$$U \equiv \bigoplus_{i=1}^{n} (|i\rangle \to U_i) \ \textit{or simply}\ U \equiv \bigoplus_{i=1}^{n} U_i$$

It is easy to check that the guarded composition U is a unitary operator in $\mathcal{H}_q \otimes \mathcal{H}$. In particular, quantum " coin" q should be considered as a system external to the principal system that has $\mathcal{H}$ as its state space; otherwise the state space of the system composed by the "coin" and the principal system is not $\mathcal{H}_q \otimes \mathcal{H}$, and defining equations (6.10) and (6.11) are inappropriate.

Actually, the guarded composition of unitary operators is nothing new; it is just a quantum multiplexor (QMUX) introduced in Subsection 2.2.4.

Example 6.3.1. *A QMUX U with k select qubits and d-qubit-wide data bus can be represented by a block-diagonal matrix:*

$$U = diag(U_0, U_1, \ldots, U_{2^k-1}) = \begin{pmatrix} U_0 & & & \\ & U_1 & & \\ & & \cdots & \\ & & & U_{2^k-1} \end{pmatrix}.$$

Multiplexing $U_0, U_1, \ldots, U_{2^k-1}$ with k select qubits is exactly the guarded composition

$$\bigoplus_{i=0}^{2^k-1} (|i\rangle \to U_i)$$

along the computational basis $\{|i\rangle\}$ of k qubits.

The guarded composition U of U_i's in Definition 6.3.1 certainly depends on the chosen orthogonal basis $\{|i\rangle\}$ of the " coin" space $\mathcal{H}_q$. For any two different orthonormal bases $\{|i\rangle\}$ and $\{|\varphi_i\rangle\}$ of the "coin space" $\mathcal{H}_q$, there exists a unitary operator U_q such that $|\varphi_i\rangle = U_q|i\rangle$ for all i. Furthermore, a routine calculation yields:

Lemma 6.3.1. *The two compositions along different bases $\{|i\rangle\}$ and $\{|\varphi_i\rangle\}$ are related to each other by*

$$\bigoplus_{i} (|\varphi_i\rangle \to U_i) = (U_q \otimes I_{\mathcal{H}}) \bigoplus_{i} (|i\rangle \to U_i)\, (U_q^\dagger \otimes I_{\mathcal{H}})$$

where $I_{\mathcal{H}}$ is the identity operator in $\mathcal{H}$.

The preceding lemma shows that the guarded composition along one orthonormal basis $\{|\varphi_i\rangle\}$ can be expressed in terms of the guarded composition along any other orthonormal basis $\{|i\rangle\}$. Therefore, the choice of orthonormal basis of the "coin space" is not essential for the definition of guarded composition.

6.3.2 OPERATOR-VALUED FUNCTIONS

A general form of guarded composition of quantum operations cannot be defined by a straightforward generalization of Definition 6.3.1. Instead, we need an auxiliary notion of operator-valued function. For any Hilbert space $\mathcal{H}$, we write $\mathcal{L}(\mathcal{H})$ for the space of (bounded linear) operators in $\mathcal{H}$.

Definition 6.3.2. *Let Δ be a nonempty set. Then a function $F : \Delta \rightarrow \mathcal{L}(\mathcal{H})$ is called an operator-valued function in $\mathcal{H}$ over Δ if*

$$\sum_{\delta\in\Delta} F(\delta)^\dagger \cdot F(\delta) \sqsubseteq I_{\mathcal{H}}, \tag{6.12}$$

where $I_{\mathcal{H}}$ is the identity operator in $\mathcal{H}$, and $\sqsubseteq$ stands for the Löwner order (see Definition 2.1.13). In particular, F is said to be full when inequality (6.12) becomes equality.

The simplest examples of operator-valued function are unitary operators and quantum measurements.

Example 6.3.2

(i) *A unitary operator U in Hilbert space $\mathcal{H}$ can be seen as a full operator-valued function over a singleton $\Delta = \{\epsilon\}$. This function maps the only element ϵ of Δ to U.*

(ii) *A quantum measurement $M = \{M_m\}$ in Hilbert space $\mathcal{H}$ can be seen as a full operator-valued function over the set $\Delta = \{m\}$ of its possible outcomes. This function maps each outcome m to the corresponding measurement operator M_m.*

It is interesting to compare part (ii) of the preceding example with Example 2.1.9 (ii) where a quantum operation is induced from a quantum measurement by ignoring the measurement outcomes. However, in the preceding example the measurement outcomes are explicitly recorded in the index set $\Delta = \{m\}$.

More generally than the preceding example, a quantum operation defines a family of operator-valued functions. Let $\mathcal{E}$ be a quantum operation in Hilbert space $\mathcal{H}$. Then $\mathcal{E}$ has the Kraus operator-sum representation:

$$\mathcal{E} = \sum_i E_i \circ E_i^\dagger,$$

meaning:

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$$

for all density operators ρ in $\mathcal{H}$ (see Theorem 2.1.1). For such a representation, we set $\Delta = \{i\}$ for the set of indexes, and define an operator-valued function over Δ by

$$F(i) = E_i$$

for every i. Since operator-sum representation of $\mathcal{E}$ is not unique, by the previous procedure $\mathcal{E}$ defines possibly more than a single operator-valued function.

Definition 6.3.3. *The set $\mathbb{F}(\mathcal{E})$ of operator-valued functions generated by a quantum operation $\mathcal{E}$ consists of the operator-valued functions defined by all different Kraus operator-sum representations of $\mathcal{E}$.*

Conversely, an operator-valued function determines uniquely a quantum operation.

Definition 6.3.4. *Let F be an operator-valued function in Hilbert space $\mathcal{H}$ over set Δ. Then F defines a quantum operation $\mathcal{E}(F)$ in $\mathcal{H}$ as follows:*

$$\mathcal{E}(F) = \sum_{\delta\in\Delta} F(\delta) \circ F(\delta)^\dagger;$$

that is,

$$\mathcal{E}(F)(\rho) = \sum_{\delta\in\Delta} F(\delta)\rho F(\delta)^\dagger$$

for every density operator ρ in $\mathcal{H}$.

To further clarify the relationship between operator-valued functions and quantum operations, for a family $\mathbb{F}$ of operator-valued functions, we write:

$$\mathcal{E}(\mathbb{F}) = \{\mathcal{E}(F) : F \in \mathbb{F}\}.$$

It is obvious that $\mathcal{E}(\mathbb{F}(\mathcal{E})) = \{\mathcal{E}\}$ for each quantum operation $\mathcal{E}$. On the other hand, for any operator-valued function F over $\Delta = \{\delta_1, \ldots, \delta_k\}$, it follows from Lemma 4.3.1 (i.e., Theorem 8.2 in [174]) that $\mathbb{F}(\mathcal{E}(F))$ consists of all operator-valued functions G over some set $\Gamma = \{\gamma_1, \ldots, \gamma_l\}$ such that

$$G(\gamma_i) = \sum_{j=1}^{n} u_{ij} \cdot F(\delta_j)$$

for each $1 \leq i \leq n$, where $n = \max(k, l)$, $U = (u_{ij})$ is an $n \times n$ unitary matrix, $F(\delta_i) = G(\gamma_j) = 0_{\mathcal{H}}$ for all $k+1 \leq i \leq n$ and $l+1 \leq j \leq n$, and $0_{\mathcal{H}}$ is the zero operator in $\mathcal{H}$.

6.3.3 GUARDED COMPOSITION OF OPERATOR-VALUED FUNCTIONS

Now we are going to define the guarded composition of operator-valued functions. Before doing so, we need to introduce a notation. Let Δ_i be a nonempty set for every $1 \leq i \leq n$. Then we write:

$$\bigoplus_{i=1}^{n} \Delta_i = \left\{\oplus_{i=1}^{n}\delta_i : \delta_i \in \Delta_i \text{ for every } 1 \leq i \leq n\right\}. \tag{6.13}$$

Here, $\oplus_{i=1}^{n}\delta_i$ is simply a notation indicating a formal, syntactic combination of δ_i $(1 \leq i \leq n)$. The intuitive meaning behind this notation will be explained when δ_i are used to denote the states of classical variables in the next section.

Definition 6.3.5. *For each $1 \leq i \leq n$, let F_i be an operator-valued function in Hilbert space $\mathcal{H}$ over set Δ_i. Let $\mathcal{H}_q$ be a "coin" Hilbert space with $\{|i\rangle\}$ as an orthonormal basis. Then the guarded composition*

$$F \stackrel{\triangle}{=} \bigoplus_{i=1}^{n}(|i\rangle \to F_i) \ \textit{or simply}\ F \stackrel{\triangle}{=} \bigoplus_{i=1}^{n} F_i$$

of F_i $(1 \leq i \leq n)$ along the basis $\{|i\rangle\}$ is an operator-valued function

$$F : \bigoplus_{i=1}^{n} \Delta_i \to \mathcal{L}(\mathcal{H}_q \otimes \mathcal{H})$$

in $\mathcal{H}_q \otimes \mathcal{H}$ over $\bigoplus_{i=1}^{n} \Delta_i$. It is defined in the following three steps:

(i) *For any $\delta_i \in \Delta_i$ $(1 \leq i \leq n)$,*

$$F(\oplus_{i=1}^{n}\delta_i)$$

is an operator in $\mathcal{H}_q \otimes \mathcal{H}$.

(ii) *For each $|\Psi\rangle \in \mathcal{H}_q \otimes \mathcal{H}$, there is a unique tuple $(|\psi_1\rangle, \dots, |\psi_n\rangle)$ such that $|\psi_1\rangle, \dots, |\psi_n\rangle \in \mathcal{H}$ and $|\Psi\rangle$ can be written as*

$$|\Psi\rangle = \sum_{i=1}^{n} |i\rangle|\psi_i\rangle,$$

and then we define

$$F(\oplus_{i=1}^{n}\delta_i)|\Psi\rangle = \sum_{i=1}^{n} \left(\prod_{k\neq i} \lambda_{k\delta_k}\right) |i\rangle(F_i(\delta_i)|\psi_i\rangle). \tag{6.14}$$

(iii) *For any $\delta_k \in \Delta_k$ $(1 \leq k \leq n)$, the coefficients*

$$\lambda_{k\delta_k} = \sqrt{\frac{trF_k(\delta_k)^\dagger F_k(\delta_k)}{\sum_{\tau_k \in \Delta_k} trF_k(\tau_k)^\dagger F_k(\tau_k)}}. \tag{6.15}$$

In particular, if F_k is full and $d = \dim \mathcal{H} < \infty$, then

$$\lambda_{k\delta_k} = \sqrt{\frac{trF_k(\delta_k)^\dagger F_k(\delta_k)}{d}}.$$

This definition is very involved. In particular, at first glance it is not easy to see where the product of $\lambda_{k\delta_k}$ in equation (6.14) comes from. A simple answer to this question is that the product is chosen for normalization of probability amplitudes. This point can be clearly seen from the proof of Lemma 6.3.3 (presented in Section 6.9 following). Intuitively, the square $\lambda_{k\delta_k}^2$ of the coefficients defined in equation (6.15) can be understood as a kind of conditional probability. Actually, some different choices of coefficients in equations (6.14) and (6.15) are possible; a further discussion on this issue is given in Subsection 6.8.1.

One thing worthy of a special attention is that the state space of guarded composition F in the preceding definition is $\mathcal{H}_q \otimes \mathcal{H}$, and thus quantum "coin" q must be treated as a system external to the principal system with state space $\mathcal{H}$.

It is easy to see that whenever Δ_i is a singleton for all $1 \le i \le n$, then all $\lambda_{k\delta_k} = 1$, and equation (6.14) degenerates to (6.11). So, the preceding definition is a generalization of guarded composition of unitary operators introduced in Definition 6.3.1.

The following lemma shows that the guarded composition of operator-valued functions is well-defined.

Lemma 6.3.2. *The guarded composition $\bigoplus_{i=1}^n (|i\rangle \to F_i)$ is an operator-valued function in $\mathcal{H}_q \otimes \mathcal{H}$ over $\bigoplus_{i=1}^n \Delta_i$. In particular, if all F_i $(1 \le i \le n)$ are full, then so is F.*

For readability, the proof of this lemma is postponed to Section 6.9. The reader is encouraged to work out a proof as an exercise.

Similar to Lemma 6.3.1, the choice of orthonormal basis of the "coin space" in the guarded composition of operator-valued function is not essential. For any two orthonormal bases $\{|i\rangle\}$ and $\{|\varphi_i\rangle\}$ of the "coin space" $\mathcal{H}_q$, let U_q be the unitary operator such that $|\varphi_i\rangle = U_q|i\rangle$ for all i. Then we have:

Lemma 6.3.3. *The two compositions along different bases $\{|i\rangle\}$ and $\{|\varphi_i\rangle\}$ are related to each other by*

$$\bigoplus_{i=1}^n (|\varphi_i\rangle \to F_i) = (U_q \otimes I_{\mathcal{H}}) \cdot \bigoplus_{i=1}^n (|i\rangle \to F_i) \cdot (U_q^\dagger \otimes I_{\mathcal{H}});$$

that is,

$$\bigoplus_{i=1}^n (|\varphi_i\rangle \to F_i)\,(\oplus_{i=1}^n \delta_i) = (U_q \otimes I_{\mathcal{H}}) \left[\bigoplus_{i=1}^n (|i\rangle \to F_i)\,(\oplus_{i=1}^n \delta_i)\right] (U_q^\dagger \otimes I_{\mathcal{H}})$$

for any $\delta_1 \in \Delta_1, \dots, \delta_n \in \Delta_n$.

We now give an example to illustrate Definition 6.3.5. This example shows how to compose two quantum measurements via a quantum " coin," which is a qubit.

Example 6.3.3. *Consider a guarded composition of two simplest quantum measurements:*

- *$M^{(0)}$ is the measurement on a qubit (the principal qubit) p in the computational basis $|0\rangle, |1\rangle$, i.e. $M^{(0)} = \{M_0^{(0)}, M_1^{(0)}\}$, where*

$$M_0^{(0)} = |0\rangle\langle 0|, \quad M_1^{(0)} = |1\rangle\langle 1|;$$

- *$M^{(1)}$ is the measurement of the same qubit but in a different basis:*

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle),$$

i.e., $M^{(1)} = \{M_+^{(1)}, M_-^{(1)}\}$, where

$$M_+^{(1)} = |+\rangle\langle+|, \quad M_-^{(1)} = |-\rangle\langle-|.$$

Then the guarded composition of $M^{(0)}$ and $M^{(1)}$ along the computational basis of another qubit (the "coin qubit") q is the measurement

$$M = M^{(0)} \oplus M^{(1)} = \{M_{0+}, M_{0-}, M_{1+}, M_{1-}\}$$

on two qubits q and p, where ij is an abbreviation of $i \oplus j$, and

$$M_{ij}(|0\rangle_q|\psi_0\rangle_p + |1\rangle_q|\psi_1\rangle_p) = \frac{1}{\sqrt{2}}\left(|0\rangle_q M_i^{(0)}|\psi_0\rangle_p + |1\rangle_q M_j^{(1)}|\psi_1\rangle_p\right)$$

for any states $|\psi_0\rangle, |\psi_1\rangle$ of the principal qubit p and $i \in \{0,1\}$, $j \in \{+,-\}$. Furthermore, for each state $|\Psi\rangle$ of two qubits q, p and for any $i \in \{0,1\}$, $j \in \{+,-\}$, a routine calculation yields that the probability that the outcome is ij when performing the guarded composition M of $M^{(0)}$ and $M^{(1)}$ on the two qubit system q, p in state $|\Psi\rangle$ is

$$p(i,j||\Psi\rangle, M) = \frac{1}{2}\left[p\left(i|_q\langle 0|\Psi\rangle, M^{(0)}\right) + p\left(j|_q\langle 1|\Psi\rangle, M^{(1)}\right)\right],$$

where:

(i) *if $|\Psi\rangle = |0\rangle_q|\psi_0\rangle_p + |1\rangle_q|\psi_1\rangle_p$, then*

$$_q\langle k|\Psi\rangle = |\psi_k\rangle$$

is the "conditional" state of the principal qubit p given that the two qubit system q, p is in state $|\Psi\rangle$ and the " coin" qubit q is in the basis state $|k\rangle$ for $k = 0, 1$;

(ii) *$p\left(i|_q\langle 0|\Psi\rangle, M^{(0)}\right)$ is the probability that the outcome is i when performing measurement $M^{(0)}$ on qubit p in state $_q\langle 0|\Psi\rangle$;*

(iii) *$p\left(j|_q\langle 1|\Psi\rangle, M^{(1)}\right)$ is the probability that the outcome is j when performing measurement $M^{(1)}$ on qubit p in state $_q\langle 1|\Psi\rangle$.*

6.3.4 GUARDED COMPOSITION OF QUANTUM OPERATIONS

In the last subsection, we learned how to compose a family of operator-valued functions employing an external quantum "coin." Now the guarded composition of a family of quantum operations can be defined through the guarded composition of the operator-valued functions generated from them.

Definition 6.3.6. *For each $1 \leq i \leq n$, let $\mathcal{E}_i$ be a quantum operation (i.e., super-operator) in Hilbert space $\mathcal{H}$. Let $\mathcal{H}_q$ be a "coin" Hilbert space with $\{|i\rangle\}$ as an orthonormal basis. Then the guarded composition of $\mathcal{E}_i$ $(1 \leq i \leq n)$ along the basis $\{|i\rangle\}$ is defined to be the family of quantum operations in $\mathcal{H}_q \otimes \mathcal{H}$:*

$$\bigoplus_{i=1}^{n} (|i\rangle \to \mathcal{E}_i) = \left\{ \mathcal{E}\left(\bigoplus_{i=1}^{n} (|i\rangle \to F_i) \right) : F_i \in \mathbb{F}(\mathcal{E}_i) \text{ for every } 1 \leq i \leq n \right\},$$

where:

(i) *$\mathbb{F}(\mathcal{F})$ stands for the set of operator-valued functions generated by quantum operation $\mathcal{F}$ (see Definition 6.3.3);*

(ii) *$\mathcal{E}(F)$ is the quantum operation defined by an operator-valued function F (see Definition 6.3.4).*

Similar to the cases in Definitions 6.3.1 and 6.3.5, the guarded composition $\bigoplus_{i=1}^{n} (|i\rangle \to \mathcal{E}_i)$ is a quantum operation in space $\mathcal{H}_q \otimes \mathcal{H}$, and thus quantum "coin" q is external to the principal system with state space $\mathcal{H}$.

It is easy to see that if $n = 1$ then the preceding guarded composition of quantum operations consists of only $\mathcal{E}_1$. For $n > 1$, however, it is usually not a singleton, as shown by the following example. For any unitary operator U in a Hilbert space $\mathcal{H}$, we write $\mathcal{E}_U = U \circ U^\dagger$ for a quantum operation defined by U, that is, $\mathcal{E}_U(\rho) = U\rho U^\dagger$ for all density operators ρ in $\mathcal{H}$ (see Example 2.1.8).

Example 6.3.4. *Suppose that U_0 and U_1 are two unitary operators in a Hilbert space $\mathcal{H}$. Let U be the composition of U_0 and U_1 guarded by the computational basis $|0\rangle, |1\rangle$ of a qubit:*

$$U = U_0 \oplus U_1.$$

Then $\mathcal{E}_U$ is an element of the guarded composition

$$\mathcal{E} = \mathcal{E}_{U_0} \oplus \mathcal{E}_{U_1}$$

of super-operators $\mathcal{E}_{U_0}$ and $\mathcal{E}_{U_1}$. But $\mathcal{E}$ contains more than one element. Indeed, it holds that

$$\mathcal{E} = \{\mathcal{E}_{U_\theta} = U_\theta \circ U_\theta^\dagger : 0 \leq \theta < 2\pi\},$$

where

$$U_\theta = U_0 \oplus e^{i\theta} U_1.$$

Note that the non-uniqueness of the members of the guarded composition $\mathcal{E}$ is caused by the relative phase θ between U_0 and U_1.

We now examine the choice of basis of the "coin space" in the guarded composition of quantum operations. To this end, we need the following two notations:

- For any two quantum operations $\mathcal{E}_1$ and $\mathcal{E}_2$ in a Hilbert space $\mathcal{H}$, their sequential composition $\mathcal{E}_2 \circ \mathcal{E}_1$ is the quantum operation in $\mathcal{H}$ defined by
$$(\mathcal{E}_2 \circ \mathcal{E}_1)(\rho) = \mathcal{E}_2(\mathcal{E}_1(\rho))$$
for any density operator ρ in $\mathcal{H}$. This notation was already introduced in Subsection 5.1.2.
- More generally, for any quantum operation $\mathcal{E}$ and any set Ω of quantum operations in Hilbert space $\mathcal{H}$, we define the sequential compositions of Ω and $\mathcal{E}$ by
$$\mathcal{E} \circ \Omega = \{\mathcal{E} \circ \mathcal{F} : \mathcal{F} \in \Omega\} \quad \text{and} \quad \Omega \circ \mathcal{E} = \{\mathcal{F} \circ \mathcal{E} : \mathcal{F} \in \Omega\}.$$

The following lemma can be easily derived from Lemma 6.3.3, and it shows that the choice of orthonormal basis of the "coin space" is not essential for the guarded composition of quantum operations. For any two orthonormal bases $\{|i\rangle\}$ and $\{|\varphi_i\rangle\}$ of the "coin space" $\mathcal{H}_q$, let U_q be the unitary operator such that $|\varphi_i\rangle = U_q|i\rangle$ for all i. Then we have:

Lemma 6.3.4. *The two compositions along different bases $\{|i\rangle\}$ and $\{|\varphi_i\rangle\}$ are related to each other by*
$$\bigoplus_{i=1}^{n}(|\varphi_i\rangle \to \mathcal{E}_i) = \left[\mathcal{E}_{U_q^\dagger \otimes I_\mathcal{H}} \circ \bigoplus_{i=1}^{n}(|i\rangle \to \mathcal{E}_i)\right] \circ \mathcal{E}_{U_q \otimes I_\mathcal{H}},$$
where $\mathcal{E}_{U_q \otimes I_\mathcal{H}}$ and $\mathcal{E}_{U_q^\dagger \otimes I_\mathcal{H}}$ are the quantum operations in $\mathcal{H}_q \otimes \mathcal{H}$ defined by unitary operators $U_q \otimes I_\mathcal{H}$ and $U_q^\dagger \otimes I_\mathcal{H}$, respectively.

Exercise 6.3.1. *Prove Lemmas 6.3.1, 6.3.3 and 6.3.4.*

6.4 SEMANTICS OF QuGCL PROGRAMS

With the preparation in Section 6.3, we are ready to define the semantics of the quantum programming language QuGCL presented in Section 6.2. Before doing it, we introduce several notations needed in this section.

- Let $\mathcal{H}$ and $\mathcal{H}'$ be two Hilbert spaces, and let E be an operator in $\mathcal{H}$. Then the cylindrical extension of E in $\mathcal{H} \otimes \mathcal{H}'$ is defined to be the operator $E \otimes I_{\mathcal{H}'}$, where $I_{\mathcal{H}'}$ is the identity operator in $\mathcal{H}'$. For simplicity, we will write E for $E \otimes I_{\mathcal{H}'}$ whenever there is no possibility of confusion.
- Let F be an operator-valued function in $\mathcal{H}$ over Δ. Then the cylindrical extension of F in $\mathcal{H} \otimes \mathcal{H}'$ is the operator-valued function $\overline{F}$ in $\mathcal{H} \otimes \mathcal{H}'$ over Δ defined by
$$\overline{F}(\delta) = F(\delta) \otimes I_{\mathcal{H}'}$$
for every $\delta \in \Delta$. For simplicity, we often write F for $\overline{F}$ whenever the context prevents any confusion.

- Let $\mathcal{E} = \sum_i E_i \circ E_i^\dagger$ be a quantum operation in $\mathcal{H}$. Then the cylindrical extension of $\mathcal{E}$ in $\mathcal{H} \otimes \mathcal{H}'$ is defined to be the quantum operation:

$$\overline{\mathcal{E}} = \sum_i (E_i \otimes I_{\mathcal{H}'}) \circ (E_i^\dagger \otimes I_{\mathcal{H}'}).$$

For simplicity, $\mathcal{E}$ will be used to denote its extension $\overline{\mathcal{E}}$ when no confusion is possible. In particular, if E is an operator in $\mathcal{H}$, and ρ is a density operator in $\mathcal{H} \otimes \mathcal{H}'$, then $E\rho E^\dagger$ should be understood as $(E \otimes I_{\mathcal{H}'})\rho(E^\dagger \otimes I_{\mathcal{H}'})$.

6.4.1 CLASSICAL STATES

The first step in defining the semantics of QuGCL is to define the states of classical variables in QuGCL. As already stated in Section 6.2, classical variables in QuGCL will only be used to record the outcomes of quantum measurements.

Definition 6.4.1. *Classical states and their domains are inductively defined as follows:*

(i) *ϵ is a classical state, called the empty state, and $dom(\epsilon) = \emptyset$;*
(ii) *If $x \in Var$ is a classical variable, and $a \in D_x$ is an element of the domain of x, then $[x \leftarrow a]$ is a classical state, and $dom([x \leftarrow a]) = \{x\}$;*
(iii) *If both δ_1 and δ_2 are classical states, and $dom(\delta_1) \cap dom(\delta_2) = \emptyset$, then $\delta_1\delta_2$ is a classical state, and $dom(\delta_1\delta_2) = dom(\delta_1) \cup dom(\delta_2)$;*
(iv) *If δ_i is a classical state for every $1 \leq i \leq n$, then $\oplus_{i=1}^n \delta_i$ is a classical state, and*

$$dom\left(\oplus_{i=1}^n \delta_i\right) = \bigcup_{i=1}^n dom(\delta_i).$$

Intuitively, a classical state δ defined by clauses (i) to (iii) in this definition can be seen as a (partial) assignment to classical variables; more precisely, δ is an element of Cartesian product $\prod_{x \in dom(\delta)} D_x$; that is, a choice function:

$$\delta : dom(\delta) \to \bigcup_{x \in dom(\delta)} D_x$$

such that $\delta(x) \in D_x$ for every $x \in dom(\delta)$. The state $\oplus_{i=1}^n \delta_i$ defined by clause (iv) is a formal combination of states δ_i $(1 \leq i \leq n)$. It will be used in defining the semantics of quantum case statement, which is a guarded composition of operator-valued functions. From equation (6.13) and Definition 6.3.5 we can see why such a combination is required. More concretely, we have:

- The empty state ϵ is the empty function. Since $\prod_{x \in \emptyset} D_x = \{\epsilon\}$, ϵ is the only possible state with an empty domain.
- The state $[x \leftarrow a]$ assigns value a to variable x but the values of the other variables are undefined.

- The composed state $\delta_1\delta_2$ can be seen as the assignment to variables in $dom(\delta_1) \cup dom(\delta_2)$ given by

$$(\delta_1\delta_2)(x) = \begin{cases} \delta_1(x) & \text{if } x \in dom(\delta_1), \\ \delta_2(x) & \text{if } x \in dom(\delta_2). \end{cases} \tag{6.16}$$

  Equation (6.16) is well-defined since it is required that $dom(\delta_1) \cap dom(\delta_2) = \emptyset$. In particular, $\epsilon\delta = \delta\epsilon = \delta$ for any state δ, and if $x \notin dom(\delta)$ then $\delta[x \leftarrow a]$ is the assignment to variables in $dom(\delta) \cup \{x\}$ given by

$$\delta[x \leftarrow a](y) = \begin{cases} \delta(y) & \text{if } y \in dom(\delta), \\ a & \text{if } y = x. \end{cases}$$

  Hence, $[x_1 \leftarrow a_1] \cdots [x_k \leftarrow a_k]$ is a classical state that assigns value a_i to variable x_i for all $1 \leq j \leq k$. It will be abbreviated to

$$[x_1 \leftarrow a_1, \cdots, x_k \leftarrow a_k]$$

  in the sequel.
- The state $\bigoplus_{i=1}^n \delta_i$ can be thought of as a kind of nondeterministic choice of δ_i $(1 \leq i \leq n)$. As will be seen in the next subsection (in particular, clause (v) of Definition 6.4.2), a classical state $\delta = [x_1 \leftarrow a_1, \cdots, x_k \leftarrow a_k]$ is actually generated by a sequence of measurements $M_1, \ldots, M_k$ with their outcomes $a_1, \ldots, a_k$ stored in variables $x_1, \ldots, x_k$, respectively. However, for quantum measurements $M_1, \ldots, M_k$, other outcomes $a'_1, \ldots, a'_k$ are possible, and then we may have many other classical states $\delta' = [x_1 \leftarrow a'_1, \cdots, x_k \leftarrow a'_k]$. So, a state of the form $\bigoplus_{i=1}^n \delta_i$ is needed to record a collection of all different outcomes of measurement sequence $M_1, \ldots, M_k$.

6.4.2 SEMI-CLASSICAL SEMANTICS

Now we can define the semi-classical semantics of QuGCL, which will serve as a stepping stone for defining its purely quantum semantics. For each QuGCL program S, we write $\Delta(S)$ for the set of all possible states of its classical variables.

- The semi-classical denotational semantics $\llbracket S \rrbracket$ of S will be defined as an operator-valued function in $\mathcal{H}_{qvar(S)}$ over $\Delta(S)$, where $\mathcal{H}_{qvar(S)}$ is the state Hilbert space of quantum variables occurring in S.

In particular, if $qvar(S) = \emptyset$, for example $S =$ **abort** or **skip**, then $\mathcal{H}_{qvar(S)}$ is a one-dimensional space $\mathcal{H}_\emptyset$, and an operator in $\mathcal{H}_\emptyset$ can be identified with a complex number; for instance, the zero operator is number 0 and the identity operator is number 1. For any set $V \subseteq qVar$ of quantum variables, we write I_V for the identity operator in Hilbert space $\mathcal{H}_V = \bigotimes_{q \in V} \mathcal{H}_q$.

Definition 6.4.2. *The classical states $\Delta(S)$ and semi-classical semantic function $\llbracket S \rrbracket$ of a QuGCL program S are inductively defined as follows:*

- **(i)** $\Delta(\mathbf{abort}) = \{\epsilon\}$, *and* $\llbracket \mathbf{abort} \rrbracket(\epsilon) = 0$;
- **(ii)** $\Delta(\mathbf{skip}) = \{\epsilon\}$, *and* $\llbracket \mathbf{skip} \rrbracket(\epsilon) = 1$;
- **(iii)** *If* $S \equiv \overline{q} := U[\overline{q}]$, *then* $\Delta(S) = \{\epsilon\}$, *and* $\llbracket S \rrbracket(\epsilon) = U_{\overline{q}}$, *where* $U_{\overline{q}}$ *is the unitary operator* U *acting in* $\mathcal{H}_{\overline{q}}$;
- **(iv)** *If* $S \equiv S_1; S_2$, *then*

$$\begin{aligned}\Delta(S) &= \Delta(S_1); \Delta(S_2)\\ &= \{\delta_1\delta_2 : \delta_1 \in \Delta(S_1), \delta_2 \in \Delta(S_2)\},\end{aligned} \tag{6.17}$$

$$\llbracket S \rrbracket(\delta_1\delta_2) = (\llbracket S_2 \rrbracket(\delta_2) \otimes I_{V \setminus qvar(S_2)}) \cdot (\llbracket S_1 \rrbracket(\delta_1) \otimes I_{V \setminus qvar(S_1)})$$

where $V = qvar(S_1) \cup qvar(S_2)$;

- **(v)** *If S is a classical case statement:*

$$S \equiv \ \mathbf{if}\ (\square m \cdot M[\overline{q} : x] = m \rightarrow S_m)\ \mathbf{fi},$$

where quantum measurement $M = \{M_m\}$, *then*

$$\Delta(S) = \bigcup_m \{\delta[x \leftarrow m] : \delta \in \Delta(S_m)\},$$

$$\llbracket S \rrbracket(\delta[x \leftarrow m]) = (\llbracket S_m \rrbracket(\delta) \otimes I_{V \setminus qvar(S_m)}) \cdot (M_m \otimes I_{V \setminus \overline{q}})$$

for every $\delta \in \Delta(S_m)$ *and for every outcome m, where*

$$V = \overline{q} \cup \left(\bigcup_m qvar(S_m) \right);$$

- **(vi)** *If S is a quantum case statement:*

$$S \equiv \ \mathbf{qif}\, [\overline{q}]\, (\square i \cdot |i\rangle \rightarrow S_i)\ \mathbf{fiq},$$

then

$$\Delta(S) = \bigoplus_i \Delta(S_i), \tag{6.18}$$

$$\llbracket S \rrbracket = \bigoplus_i (|i\rangle \rightarrow \llbracket S_i \rrbracket), \tag{6.19}$$

where operation $\bigoplus$ *in equation (6.18) is defined by equation (6.13), and* $\bigoplus$ *in equation (6.19) stands for the guarded composition of operator-valued functions (see Definition 6.3.5).*

Since it is required in Definition 6.2.1 that $var(S_1) \cap var(S_2) = \emptyset$ in the sequential composition $S_1; S_2$, we have $dom(\delta_1) \cap dom(\delta_2) = \emptyset$ for any $\delta_1 \in \Delta(S_1)$ and $\delta_2 \in \Delta(S_2)$. Thus, equation (6.17) is well-defined.

Intuitively, the semi-classical semantics of quantum programs can be imagined as follows:

- If a quantum program S does not contain any quantum case statement, then its semantic structure is a tree with its nodes labelled by basic commands and its edges by linear operators. This tree grows up from the root in the following way:
  - if the current node is labelled by a unitary transformation U, then a single edge stems from the node and it is labelled by U; and
  - if the current node is labelled by a measurement $M = \{M_m\}$, then for each possible outcome m, an edge stems from the node and it is labelled by the corresponding measurement operator M_m.

  Obviously, branching in the semantic tree comes from the different possible outcomes of a measurement in S. Each classical state $\delta \in \Delta(S)$ is corresponding to a branch in the semantic tree of S, and it denotes a possible path of execution. Furthermore, the value of semantic function $\llfloor S \rrfloor$ in state δ is the (sequential) composition of the operators labelling the edges of δ. This can be clearly seen from clauses (i) - (v) of the preceding definition.
- The semantic structure of a quantum program S with quantum case statements is much more complicated. It can be seen as a tree with superpositions of nodes that generate superpositions of branches. The value of semantic function $\llfloor S \rrfloor$ in a superposition of branches is then defined as the guarded composition of the values in these branches.

6.4.3 PURELY QUANTUM SEMANTICS

The purely quantum semantics of a quantum program written in QuGCL can be naturally defined as the quantum operation induced by its semi-classical semantic function (see Definition 6.3.4).

Definition 6.4.3. *For each QuGCL program S, its purely quantum denotational semantics is the quantum operation $[\![S]\!]$ in $\mathcal{H}_{qvar(S)}$ defined as follows:*

$$[\![S]\!] = \mathcal{E}(\llfloor S \rrfloor) = \sum_{\delta \in \Delta(S)} \llfloor S \rrfloor(\delta) \circ \llfloor S \rrfloor(\delta)^\dagger, \tag{6.20}$$

where $\llfloor S \rrfloor$ is the semi-classical semantic function of S.

The following proposition presents an explicit representation of the purely quantum semantics of a program in terms of its subprograms. This representation is easier to use in applications than the preceding abstract definition.

Proposition 6.4.1

(i) $[\![\mathbf{abort}]\!] = 0$;
(ii) $[\![\mathbf{skip}]\!] = 1$;
(iii) $[\![S_1; S_2]\!] = [\![S_2]\!] \circ [\![S_1]\!]$;

(iv) $[\![\overline{q} := U[\overline{q}]]\!] = U_{\overline{q}} \circ U_{\overline{q}}^{\dagger}$;

(v)

$$[\![\mathbf{if}\ (\square m \cdot M[\overline{q} : x] = m \to S_m)\ \mathbf{fi}]\!] = \sum_m \left[[\![S_m]\!] \circ (M_m \circ M_m^{\dagger}) \right].$$

Here, $[\![S_m]\!]$ should be seen as a cylindrical extension in $\mathcal{H}_V$ from $\mathcal{H}_{qvar(S_m)}$, $M_m \circ M_m^{\dagger}$ is seen as a cylindrical extension in $\mathcal{H}_V$ from $\mathcal{H}_{\overline{q}}$, and

$$V = \overline{q} \cup \left(\bigcup_m qvar(S_m) \right);$$

(vi)

$$[\![\mathbf{qif}\ [\overline{q}]\ (\square i \cdot |i\rangle \to S_i)\ \mathbf{fiq}]\!] \in \bigoplus_i (|i\rangle \to [\![S_i]\!]). \tag{6.21}$$

Here $[\![S_i]\!]$ should be understood as a cylindrical extension in $\mathcal{H}_V$ from $\mathcal{H}_{qvar(S_i)}$ for every $1 \leq i \leq n$, and

$$V = \overline{q} \cup \left(\bigcup_i qvar(S_i) \right).$$

It should be mentioned that symbol $\circ$ in clause (iii) and its first occurrence in clause (v) of the preceding proposition stands for composition of quantum operations; that is, $(\mathcal{E}_2 \circ \mathcal{E}_1)(\rho) = \mathcal{E}_2(\mathcal{E}_1(\rho))$ for all density operators ρ. But the symbol $\circ$ in clause (iv) and its second occurrence in clause (v) is used to define a quantum operation from an operator; that is, for an operator A, $A \circ A^{\dagger}$ is the quantum operation $\mathcal{E}_A$ defined by $\mathcal{E}_A(\rho) = A\rho A^{\dagger}$ for every density operator ρ. Essentially, clauses (ii) - (v) in this proposition are the same as the corresponding clauses in Proposition 3.3.1. The proof of this proposition is deferred to Section 6.9. Actually, the proof is not difficult, although it is tedious. The reader is encouraged to try to prove this proposition in order to gain a better understanding of Definitions 6.4.2 and 6.4.3.

The preceding proposition shows that the purely quantum denotational semantics is *almost compositional*, but it is *not completely compositional* because the symbol "$\in$" appears in equation (6.21). The symbol "$\in$" can be understood as a *refinement relation*. It is worth noting that in general the symbol "$\in$" in (6.21) cannot be replaced by equality. This is exactly the reason that the purely quantum semantics of a program has to be derived through its semi-classical semantics but cannot be defined directly by a structural induction.

It should be stressed that the symbol "$\in$" in equation (6.21) does not mean that the purely quantum semantics of the quantum case statement is not well-defined. In fact, it is uniquely defined by equations (6.19) and (6.20) as a quantum operation. The right-hand side of equation (6.21) is not the semantics of any program. It is the guarded composition of the semantics of programs S_i. Since it is the guarded composition of a family of quantum operations, it can be a set consisting of more than one quantum operation, as shown in Example 6.3.4. The semantics of the quantum case statement is one member of the set of quantum operations in the right-hand side of equation (6.21).

Exercise 6.4.1. *Find an example to show that equation (6.21) is not true when the symbol "$\in$" is replaced by equality.*

Equivalence between quantum programs can be introduced based on their purely quantum denotational semantics. Roughly speaking, two programs are equivalent if the outputs computed by them are the same for the same input. Formally, we have:

Definition 6.4.4. *Let P and Q be two QuGCL programs. Then:*

(i) *We say that P and Q are equivalent and write $P = Q$ if*

$$[\![P]\!] \otimes \mathcal{I}_{Q\setminus P} = [\![Q]\!] \otimes \mathcal{I}_{P\setminus Q},$$

where $\mathcal{I}_{Q\setminus P}$ is the identity quantum operation in $\mathcal{H}_{qvar(Q)\setminus qvar(P)}$ and $\mathcal{I}_{P\setminus Q}$ the identity quantum operation in $\mathcal{H}_{qvar(P)\setminus qvar(Q)}$.

(ii) *The "coin-free" equivalence $P =_{CF} Q$ holds if*

$$tr_{\mathcal{H}_{cvar(P)\cup cvar(Q)}}([\![P]\!] \otimes \mathcal{I}_{Q\setminus P}) = tr_{\mathcal{H}_{cvar(P)\cup cvar(Q)}}([\![Q]\!] \otimes \mathcal{I}_{P\setminus Q}).$$

The symbol "tr" in this equation denotes partial trace. The partial trace on density operators was introduced in Definition 2.1.22. Furthermore, the notion of partial trace can be generalized to quantum operations: for any quantum operation $\mathcal{E}$ in $\mathcal{H}_1 \otimes \mathcal{H}_2$, $tr_{\mathcal{H}_1}(\mathcal{E})$ is a quantum operation from $\mathcal{H}_1 \otimes \mathcal{H}_2$ to $\mathcal{H}_2$ defined by

$$tr_{\mathcal{H}_1}(\mathcal{E})(\rho) = tr_{\mathcal{H}_1}(\mathcal{E}(\rho))$$

for all density operators ρ in $\mathcal{H}_1 \otimes \mathcal{H}_2$.

Obviously, $P = Q$ implies $P =_{CF} Q$. The "coin-free" equivalence means that "coin" variables are only used to produce quantum control flows of programs (or to realize superposition of programs, as discussed in the next section). The computational outcome of a program P is stored in the "principal" state space $\mathcal{H}_{qvar(P)\setminus cvar(P)}$. For the special case of $qvar(P) = qvar(Q)$, we have:

- $P = Q$ if and only if $[\![P]\!] = [\![Q]\!]$; and
- $P =_{CF} Q$ if and only if $tr_{\mathcal{H}_{cvar(P)}}[\![P]\!] = tr_{\mathcal{H}_{cvar(P)}}[\![Q]\!]$.

The notions of equivalence given in the previous definition provide a basis for quantum program transformation and optimization where the transformed program is required to be equivalent to the source program. A set of algebraic laws that can help to establish equivalence between QuGCL programs will be presented in Section 6.6.

6.4.4 WEAKEST PRECONDITION SEMANTICS

The notion of quantum weakest precondition was introduced in Subsection 4.1.1, and the weakest precondition semantics of quantum **while**-programs was presented in Subsection 4.2.2. Here, we give the weakest precondition semantics of QuGCL programs, which can be derived from Proposition 6.4.1 together with Proposition 4.1.1.

As in classical programming and in the case of quantum **while**-programs, weakest precondition semantics provides us with a way for analyzing QuGCL programs backwards.

Proposition 6.4.2

(i) $wp.\mathbf{abort} = 0$;
(ii) $wp.\mathbf{skip} = 1$;
(iii) $wp.(P_1; P_2) = wp.P_2 \circ wp.P_1$;
(iv) $wp.\overline{q} := U[\overline{q}] = U^{\dagger}_{\overline{q}} \circ U_{\overline{q}}$;
(v)

$$wp.\mathbf{if}\ (\square m \cdot M[\overline{q} : x] = m \rightarrow P_m)\ \mathbf{fi}$$
$$= \sum_m \left[(M^{\dagger}_m \circ M_m) \circ wp.P_m\right];$$

(vi) $wp.\mathbf{qif}\,[\,\overline{q}\,]\,(\square i \cdot |i\rangle \rightarrow P_i)\ \mathbf{fiq} \in \square_i\,(|i\rangle \rightarrow wp.P_i)$.

Some cylindrical extensions of quantum operations are used but unspecified in the preceding proposition because they can be recognized from the context. It should be noticed that the symbol $\circ$ is used in two different ways, as remarked after Proposition 6.4.1. Again, the symbol "$\in$" in the preceding clause (vi) cannot be replaced by equality because the right-hand side of clause (vi) is a set that may contain more than one quantum operation.

We can define the refinement relation between quantum QuGCL programs in terms of their weakest precondition semantics. To this end, we first generalize the Löwner order to the case of quantum operations: for any two quantum operations $\mathcal{E}$ and $\mathcal{F}$ in Hilbert space $\mathcal{H}$,

- $\mathcal{E} \sqsubseteq \mathcal{F}$ if and only if $\mathcal{E}(\rho) \sqsubseteq \mathcal{F}(\rho)$ for all density operators ρ in $\mathcal{H}$.

Definition 6.4.5. *Let P and Q be two QuGCL programs. Then we say that P is refined by Q and write $P \sqsubseteq Q$ if*

$$wp.P \otimes \mathcal{I}_{Q\backslash P} \sqsubseteq wp.Q \otimes \mathcal{I}_{P\backslash Q},$$

where $\mathcal{I}_{Q\backslash P}$ and $\mathcal{I}_{P\backslash Q}$ are the same as in Definition 6.4.4.

Intuitively, $P \sqsubseteq Q$ means that P is improved by Q because the precondition of P is weakened to the precondition of Q. It is easy to see that $P \sqsubseteq Q$ and $Q \sqsubseteq P$ implies $P \equiv Q$. The notion of "coin-free" refinement can be defined in a way similar to Definition 6.4.4 (ii).

The refinement techniques have been successfully developed in classical programming so that specifications (of users' requirements) can be refined step by step using various refinement laws and finally transferred to codes that can be executed on machines; see [27] and [172] for a systematic exposition of refinement techniques. These techniques were also extended to probabilistic programming in [220]. Here, we are not going to further consider how refinement techniques can be used in quantum programming, but leave it as a topic for future research.

6.4.5 AN EXAMPLE

To close out this section, we present a simple example that helps us to understand the semantic notions introduced here.

Example 6.4.1. *Let q be a qubit variable and x, y two classical variables. Consider the QuGCL program*

```
P ≡ qif |0⟩ → H[q];
               if M^(0)[q : x] = 0 → X[q];
               □                 1 → Y[q]
               fi
        □ |1⟩ → S[q];
               if M^(1)[q : x] = 0 → Y[q]
               □                 1 → Z[q]
               fi;
               X[q];
               if M^(0)[q : y] = 0 → Z[q]
               □                 1 → X[q]
               fi
    fiq
```

where $M^{(0)}, M^{(1)}$ are the measurements on a qubit in computational basis $|0\rangle, |1\rangle$ and basis $|\pm\rangle$, respectively (see Example 6.3.3), H is the Hadamard gate, X, Y, Z are the Pauli matrices, and S is the phase gate (see Examples 2.2.1 and 2.2.2). The program P is a quantum case statement between two subprograms P_0 and P_1 with the "coin" omitted. The first subprogram P_0 is the Hadamard gate followed by the measurement in the computational basis, where whenever the outcome is 0, then the gate X follows; whenever the outcome is 1, then the gate Y follows. The second subprogram P_1 is the gate S followed by the measurement in basis $|\pm\rangle$, the gate X, and the measurement in the computational basis.

For simplicity, we write a for classical state $[x \leftarrow a]$ of program P_0 and bc for classical state $[x \leftarrow b, y \leftarrow c]$ of program P_1 for any $a, c \in \{0, 1\}$ and $b \in \{+, -\}$. Then the semi-classical semantic functions of P_0 and P_1 are given as follows:

$$\begin{cases} [\![P_0]\!](0) = X \cdot |0\rangle\langle 0| \cdot H = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, \\ [\![P_0]\!](1) = Y \cdot |1\rangle\langle 1| \cdot H = \frac{i}{\sqrt{2}}\begin{pmatrix} -1 & 1 \\ 0 & 0 \end{pmatrix}, \\ [\![P_1]\!](+0) = Z \cdot |0\rangle\langle 0| \cdot X \cdot Y \cdot |+\rangle\langle +| \cdot S = \frac{1}{2}\begin{pmatrix} i & -1 \\ 0 & 0 \end{pmatrix}, \end{cases}$$

$$\begin{cases} [\![P_1]\!](+1) = X \cdot |1\rangle\langle 1| \cdot X \cdot Y \cdot |+\rangle\langle +| \cdot S = \frac{1}{2}\begin{pmatrix} -i & 1 \\ 0 & 0 \end{pmatrix}, \\ [\![P_1]\!](-0) = Z \cdot |0\rangle\langle 0| \cdot X \cdot Z \cdot |-\rangle\langle -| \cdot S = \frac{1}{2}\begin{pmatrix} 1 & -i \\ 0 & 0 \end{pmatrix}, \\ [\![P_1]\!](-1) = X \cdot |1\rangle\langle 1| \cdot X \cdot Z \cdot |-\rangle\langle -| \cdot S = \frac{1}{2}\begin{pmatrix} 1 & -i \\ 0 & 0 \end{pmatrix}. \end{cases}$$

The semi-classical semantic function of P is an operator-valued function in the state space of two qubits over classical states

$$\Delta(P) = \{a \oplus bc : a, c \in \{0,1\} \textit{ and } b \in \{+,-\}\}.$$

It follows from equation (6.14) that

$$[\![P]\!](a \oplus bc)(|0\rangle|\varphi\rangle) = \lambda_{1(bc)}|0\rangle([\![P_0]\!](a)|\varphi\rangle),$$
$$[\![P]\!](a \oplus bc)(|1\rangle|\varphi\rangle) = \lambda_{0a}|1\rangle([\![P_1]\!](bc)|\varphi\rangle),$$

where $\lambda_{0a} = \frac{1}{\sqrt{2}}$ *and* $\lambda_{1(bc)} = \frac{1}{2}$ *for* $a, c \in \{0,1\}$ *and* $b \in \{+,-\}$. *Using*

$$[\![P]\!](a \oplus bc) = \sum_{i,j \in 0,1} ([\![P]\!](a \oplus bc)|ij\rangle)\langle ij|,$$

we can compute:

$$[\![P]\!](0 \oplus +0) = \frac{1}{2\sqrt{2}}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix},$$

$$[\![P]\!](0 \oplus +1) = \frac{1}{2\sqrt{2}}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$[\![P]\!](0 \oplus -0) = [\![P]\!](0 \oplus -1) = \frac{1}{2\sqrt{2}}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -i & 0 \end{pmatrix},$$

$$[\![P]\!](1 \oplus +0) = \frac{1}{2\sqrt{2}}\begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix},$$

$$\llbracket P \rrbracket (1 \oplus +1) = \frac{1}{2\sqrt{2}} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\llbracket P \rrbracket (1 \oplus -0) = \llbracket P \rrbracket (1 \oplus -1) = \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -i & 0 \end{pmatrix}.$$

Then the purely quantum semantics of program P is the quantum operation:

$$\llbracket P \rrbracket = \sum_{a,c\in\{0,1\} \text{ and } b\in\{+,-\}} E_{abc} \circ E_{abc}^{\dagger},$$

where $E_{abc} = \llbracket P \rrbracket (a \oplus bc)$. Moreover, it follows from Proposition 4.1.1 that the weakest precondition semantics of P is the quantum operation:

$$wp.P = \sum_{a,c\in\{0,1\} \text{ and } b\in\{+,-\}} E_{abc}^{\dagger} \circ E_{abc}.$$

Exercise 6.4.2. *Use the preceding example to convince yourself that equation (6.6) is not suitable for defining the semantics of a quantum case statement of which some branch contains measurements.*

6.5 QUANTUM CHOICE

In the previous three sections, we introduced the syntax and semantics of the quantum programming language QuGCL with the new program construct of quantum case statement. A notion of quantum choice can be defined in terms of the quantum case statement. This notion is very useful for simplification of the presentation. But more importantly, it is of independent significance conceptually.

6.5.1 CHOICES: FROM CLASSICAL TO QUANTUM via PROBABILISTIC

The initial idea of quantum choice also comes from the definition of quantum walks. To motivate the notion of quantum choice, let us go through a conceptual transition from nondeterministic choice to probabilistic choice and then to quantum choice.

(i) **Classical Choice**: We first observe that nondeterminism arises from case statement (6.2) as a consequence of the " overlapping" of the guards $G_1, G_2, \ldots, G_n$; that is, if more than one guards G_i are true at the same time, the case statement needs to select one from the corresponding commands S_i for execution. In particular, if $G_1 = G_2 = \cdots = G_n = \textbf{true}$, then the case statement becomes a demonic choice:

$$\square_{i=1}^{n} S_i \tag{6.22}$$

where the alternatives S_i are chosen unpredictably.

(ii) **Probabilistic Choice**: To formalize randomized algorithms, research on probabilistic programming started in the 1980s with the introduction of probabilistic choice:

$$\square_{i=1}^{n} S_i@p_i \tag{6.23}$$

where $\{p_i\}$ is a probability distribution; that is, $p_i \geq 0$ for all i, and $\sum_{i=1}^{n} p_i = 1$. The probabilistic choice (6.23) randomly chooses the command S_i with probability p_i for every i, and thus it can be seen as a refinement (or resolution) of the demonic choice (6.22).

(iii) **Quantum Choice**: Recall from Examples 2.3.1 and 2.3.2 that the single-step operator of a quantum walk is a " coin tossing operator" followed by a shift operator, which, as indicated in Section 6.1, can be seen as a quantum case statement. Simply following this idea, a general form of quantum choice can be easily defined based on the notion of quantum case statement.

Definition 6.5.1. *Let S be a program such that $\overline{q} = qvar(S)$, and let S_i be programs for all i. Assume that quantum variables $\overline{q}$ are external to all S_i; that is,*

$$\overline{q} \cap \left(\bigcup_i qvar(S_i)\right) = \emptyset.$$

If $\{|i\rangle\}$ is an orthonormal basis of $\mathcal{H}_{\overline{q}}$, the state Hilbert space of the "coin" system denoted by $\overline{q}$, then the quantum choice of S_i's with " coin-tossing" program S along the basis $\{|i\rangle\}$ is defined as

$$[S]\left(\bigoplus_i |i\rangle \rightarrow S_i\right) \triangleq S; \textbf{qif}\ [\overline{q}]\ (\square i \cdot |i\rangle \rightarrow S_i)\ \textbf{fiq}. \tag{6.24}$$

In particular, if $n = 2$, then the quantum choice will be abbreviated to $S_0\ {}_S\!\oplus S1$ or $S_0 \oplus\ {}_S S_1$.

This definition is not easy to understand in an abstract way. For a better understanding of it, the reader should revisit Examples 2.3.1 and 2.3.2 with the idea of this definition in mind. At this point, she/he can also move to read Example 6.7.1 following.

Since a quantum choice is defined in terms of quantum case statements, the semantics of the former can be directly derived from that of the latter.

Obviously, if the "coin-tossing program" S does nothing, that is, its semantics is the identity operator in $\mathcal{H}_{\overline{q}}$, for example $S = \textbf{skip}$, then quantum choice "$[S]\left(\bigoplus_i |i\rangle \rightarrow S_i\right)$" coincides with quantum case statement "$\textbf{qif}\ [\overline{q}]\ (\square i \cdot |i\rangle \rightarrow S_i)$ $\textbf{fiq}$". In general, however, we should carefully distinguish a quantum choice from a quantum case statement.

It is interesting to compare quantum choice (6.24) with probabilistic choice (6.23). As said before, a probabilistic choice is a resolution of nondeterminism. In a probabilistic choice, we can simply say that the choice is made according to a certain probability distribution, and do not have necessarily to specify how this distribution is generated. However, when defining a quantum choice, a "device" that can actually perform the choice, namely a "quantum coin," has to be explicitly introduced. So, a quantum choice can be further seen as a resolution of nondeterminism in the choice of (quantum) "devices" that generate the probability distribution of a probabilistic choice. A mathematical formulation of this idea will be given in the next subsection.

A Quantum Programming Paradigm – Superposition-of-Programs:

A *programming paradigm* is a way of building the structure and elements of programs. A programming language with quantum case statement and quantum choice supports a new quantum programming paradigm – *superposition-of-programs*. The basic idea of superposition-of-programs was already briefly discussed in Subsection 1.2.2. Now, after introducing the formal definition of quantum choice, this idea becomes much clearer. Actually, programmers can think of quantum choice (6.24) as a superposition-of-programs. More precisely, quantum choice (6.24) first runs " coin-tossing" program S to create a superposition of the respective execution paths of programs S_i $(1 \leq i \leq n)$, and then enters a quantum case statement of $S_1, \ldots, S_n$. During the execution of the quantum case statement, each S_i is running along its own path within the whole superposition of execution paths of $S_1, \ldots, S_n$.

Superposition-of-programs can be thought of as a higher-level superposition than *superposition-of-data*. The idea of superposition-of-data is well-understood by the quantum computation community, and the studies of quantum programming in the previous chapters have been carried out around this idea. However, the studies of superposition-of-programs are still at the very beginning, and this chapter and the next represent the first step toward this new quantum programming paradigm. Quantum case statement and quantum choice are two important ingredients in the realization of the quantum programming paradigm of superposition-of-programs. But only quantum case statement was introduced as a primitive program construct in the syntax of QuGCL because quantum choice may be easily defined as a derived program construct from quantum case statement. In the next chapter, the notion of quantum recursion with quantum control flow will be introduced in order to further realize the superposition-of-programs paradigm.

6.5.2 QUANTUM IMPLEMENTATION OF PROBABILISTIC CHOICE

The relationship between probabilistic choice and quantum choice was briefly discussed after Definition 6.5.1. Now we examine this relationship in a more precise way. To this end, we first expand the syntax and semantics of QuGCL to include probabilistic choice.

Definition 6.5.2. *Let P_i be a QuGCL program for each $1 \leq i \leq n$, and let $\{p_i\}_{i=1}^n$ be a sub-probability distribution; that is, $p_i > 0$ for each $1 \leq i \leq n$ and $\sum_{i=1}^n p_i \leq 1$. Then*

(i) *The probabilistic choice of $P_1, \ldots, P_n$ according to $\{p_i\}_{i=1}^n$ is*

$$\sum_{i=1}^{n} P_i @ p_i.$$

(ii) *The quantum variables of the choice are:*

$$qvar\left(\sum_{i=1}^{n} P_i@p_i\right) = \bigcup_{i=1}^{n} qvar(P_i).$$

(iii) *The purely quantum denotational semantics of the choice is:*

$$\left[\!\!\left[\sum_{i=1}^{n} P_i@p_i\right]\!\!\right] = \sum_{i=1}^{n} p_i \cdot [\![P_i]\!]. \tag{6.25}$$

Intuitively, program $\sum_{i=1}^{n} P_i@p_i$ chooses P_i to execute with probability p_i for every $1 \leq i \leq n$, and it aborts with probability $1 - \sum_{i=1}^{n} p_i$. The right-hand side of equation (6.25) is the probabilistic combination of quantum operations $[\![P_i]\!]$ according to distribution $\{p_i\}$; that is,

$$\left(\sum_{i=1}^{n} p_i \cdot [\![P_i]\!]\right)(\rho) = \sum_{i=1}^{n} p_i \cdot [\![P_i]\!](\rho)$$

for all density operators ρ. It is obvious that $\sum_{i=1}^{n} p_i \cdot [\![P_i]\!]$ is a quantum operation too.

A clear description about the relationship between probabilistic choice and quantum choice requires us to further expand the syntax and semantics of QuGCL by introducing local quantum variables.

Definition 6.5.3. *Let S be a QuGCL program, $\overline{q}$ a quantum register and ρ a density operator in $\mathcal{H}_{\overline{q}}$. Then*

(i) *The block command defined by S restricted to $\overline{q} = \rho$ is:*

$$\textbf{begin local } \overline{q} := \rho; S \textbf{ end}. \tag{6.26}$$

(ii) *The quantum variables of the block command are:*

$$qvar\,(\textbf{begin local } \overline{q} := \rho; S \textbf{ end}) = qvar(S) \setminus \overline{q}.$$

(iii) *The purely quantum denotational semantics of the block command is given as follows:*

$$[\![\textbf{begin local } \overline{q} := \rho; S \textbf{ end}]\!]\,(\sigma) = tr_{\mathcal{H}_{\overline{q}}}([\![S]\!](\sigma \otimes \rho))$$

for any density operator σ in $\mathcal{H}_{qvar(S)\setminus\overline{q}}$. Here, the symbol "tr" stands for partial trace (see Definition 2.1.22).

This definition is essentially a restatement of Definition 3.3.7. The only difference between them is that in block (6.26), $\overline{q}$ has to be initialized before program S using the statement $\overline{q} := \rho$, since initialization is not included in the syntax of QuGCL.

Let us consider a simple example that can help us to understand the preceding two definitions.

Example 6.5.1. (Continuation of Example 6.3.3; Probabilistic mixture of measurements). *Let $M^{(0)}$ and $M^{(1)}$ be the measurements on a qubit in the computational basis and in the basis $|\pm\rangle$, respectively. We consider a random choice between $M^{(0)}$ and $M^{(1)}$.*

- *If we perform measurement $M^{(0)}$ on qubit p in state $|\psi\rangle$ and discard the outcomes of measurement, then we get*

$$\rho_0 = M_0^{(0)}|\psi\rangle\langle\psi|M_0^{(0)} + M_1^{(0)}|\psi\rangle\langle\psi|M_1^{(0)};$$

- *If we perform measurement $M^{(1)}$ on $|\psi\rangle$ and discard the outcomes, then we get*

$$\rho_1 = M_+^{(1)}|\psi\rangle\langle\psi|M_+^{(1)} + M_-^{(1)}|\psi\rangle\langle\psi|M_-^{(1)}.$$

Here, measurement operators $M_0^{(0)}, M_1^{(0)}, M_+^{(1)}, M_-^{(1)}$ are as in Example 6.3.3. We now take the unitary matrix

$$U = \begin{pmatrix} \sqrt{s} & \sqrt{r} \\ \sqrt{r} & -\sqrt{s} \end{pmatrix}$$

where $s, r \geq 0$ and $s + r = 1$, and introduce a "coin" qubit q. Let

$$\begin{array}{lll} P_i \equiv \textbf{if}\ M^{(i)}[p:x] = & 0 \rightarrow \textbf{skip} \\ \quad\ \ \square & 1 \rightarrow \textbf{skip} \\ \quad\ \ \textbf{fi} & \end{array}$$

for $i = 0, 1$, and put the quantum choice of P_0 and P_1 according to the "coin tossing operator" U into a block with the "coin" qubit q as a local variable:

$$P \equiv \textbf{ begin local } q := |0\rangle; P_0\ {}_{U[q]}\!\oplus P_1 \textbf{ end}$$

Then for any $|\psi\rangle \in \mathcal{H}_p$, $i \in \{0,1\}$ and $j \in \{+,-\}$, we have:

$$\begin{aligned} [\![P]\!](|\psi\rangle\langle\psi|) &= tr_{\mathcal{H}_q}\left(\sum_{i\in\{0,1\}\ \text{and}\ j\in\{+,-\}} |\psi_{ij}\rangle\langle\psi_{ij}|\right) \\ &= 2\left(\sum_{i\in\{0,1\}} \frac{s}{2} M_i^{(0)}|\psi\rangle\langle\psi|M_i^{(0)} + \sum_{j\in\{+,-\}} \frac{r}{2} M_j^{(1)}|\psi\rangle\langle\psi|M_j^{(1)}\right) \\ &= s\rho_0 + r\rho_1, \end{aligned}$$

where:

$$|\psi_{ij}\rangle \triangleq M_{ij}(U|0\rangle|\psi\rangle) = \sqrt{\frac{s}{2}}|0\rangle M_i^{(0)}|\psi\rangle + \sqrt{\frac{r}{2}}|1\rangle M_j^{(1)}|\psi\rangle,$$

and measurement operators M_{ij} are as in Example 6.3.3. So, program P can be seen as a probabilistic mixture of measurements $M^{(0)}$ and $M^{(1)}$, with respective probabilities s, r.

Now we are ready to precisely characterize the relationship between probabilistic choice and quantum choice. Roughly speaking, if the “coin” variables are immediately treated as local variables, then a quantum choice degenerates to a probabilistic choice.

Theorem 6.5.1. *Let $qvar(S) = \overline{q}$. Then we have:*

$$\textbf{begin local } \overline{q} := \rho; [S] \left(\bigoplus_{i=1}^{n} |i\rangle \rightarrow S_i \right) \textbf{ end} = \sum_{i=1}^{n} S_i @ p_i \tag{6.27}$$

where probability $p_i = \langle i | [\![S]\!](\rho) | i \rangle$ for every $1 \leq i \leq n$.

For readability, the tedious proof of this theorem is deferred to Section 6.9. The inverse of this theorem is also true. For any probability distribution $\{p_i\}_{i=1}^{n}$, we can find an $n \times n$ unitary operator U such that

$$p_i = |U_{i0}|^2 \ (1 \leq i \leq n).$$

So, it follows immediately from the preceding theorem that a probabilistic choice $\sum_{i=1}^{n} S_i @ p_i$ can always be implemented by a quantum choice:

$$\textbf{begin local } \overline{q} := |0\rangle; [U[\overline{q}]] \left(\bigoplus_{i=1}^{n} |i\rangle \rightarrow S_i \right) \textbf{ end}$$

where $\overline{q}$ is a family of new quantum variables with an n-dimensional state space. As said in Subsection 6.5.1, probabilistic choice (6.23) can be thought of as a refinement of nondeterministic choice (6.22). Since for a given probability distribution $\{p_i\}$, there is more than one “coin program” S to implement the probabilistic choice $\sum_{i=1}^{n} S_i @ p_i$ in equation (6.27), a quantum choice can be further seen as a refinement of a probabilistic choice where a specific “device” (quantum “coin”) is explicitly given for generating the probability distribution $\{p_i\}$.

6.6 ALGEBRAIC LAWS

The algebraic approach has been employed in classical programming, which establishes various algebraic laws for programs so that calculation of programs becomes possible by using these laws. In particular, algebraic laws are useful for verification, transformation and compilation of programs. In this section, we present a family of basic algebraic laws for quantum case statement and quantum choice. For readability, all of the proofs of these laws are postponed to Section 6.9.

The laws given in the following theorem show that the quantum case statement is idempotent, commutative and associative, and sequential composition is distributive over the quantum case statement from the right.

Theorem 6.6.1 (Laws for Quantum Case Statement).

(i) *Idempotent Law: If $S_i = S$ for all i, then*

$$\mathbf{qif}\ (\square i \cdot |i\rangle \to S_i)\ \mathbf{fiq} = S.$$

(ii) *Commutative Law: For any permutation τ of $\{1,\ldots,n\}$, we have:*

$$\begin{aligned}&\mathbf{qif}\,[\overline{q}\,]\left(\square_{i=1}^{n} i \cdot |i\rangle \to S_{\tau(i)}\right)\ \mathbf{fiq}\\ &\qquad = U_{\tau^{-1}}[\overline{q}\,]; \mathbf{qif}\,[\overline{q}\,]\left(\square_{i=1}^{n} i \cdot |i\rangle \to S_i\right)\ \mathbf{fiq}; U_\tau[\overline{q}\,],\end{aligned}$$

where:

(a) *τ^{-1} is the inverse of τ, i.e., $\tau^{-1}(i) = j$ if and only if $\tau(j) = i$ for $i,j \in \{1,\ldots,n\}$; and*

(b) *U_τ (respectively $U_{\tau^{-1}}$) is the unitary operator permutating the basis $\{|i\rangle\}$ of $\mathcal{H}_{\overline{q}}$ with τ (respectively τ^{-1}); that is,*

$$U_\tau(|i\rangle) = |\tau(i)\rangle\ (\textit{respectively}\ U_{\tau^{-1}}(|i\rangle) = |\tau^{-1}(i)\rangle)$$

for every $1 \le i \le n$.

(iii) *Associative Law:*

$$\begin{aligned}&\mathbf{qif}\ \left(\square i \cdot |i\rangle \to \mathbf{qif}\ \left(\square j_i \cdot |j_i\rangle \to S_{ij_i}\right)\ \mathbf{fiq}\right)\ \mathbf{fiq}\\ &\qquad = \mathbf{qif}\ (\overline{\alpha})\left(\square i,j_i \cdot |i,j_i\rangle \to S_{ij_i}\right)\ \mathbf{fiq}\end{aligned}$$

for some family $\overline{\alpha}$ of parameters, where the right-hand side is a parameterized quantum case statement that will be defined in Subsection 6.8.1.

(iv) *Distributive Law: If $\overline{q} \cap qvar(Q) = \emptyset$, then*

$$\mathbf{qif}\,[\overline{q}\,]\,(\square i \cdot |i\rangle \to S_i)\ \mathbf{fiq}; Q =_{CF} \mathbf{qif}\ (\overline{\alpha})[\overline{q}\,]\,(\square i \cdot |i\rangle \to (S_i; Q))\,\mathbf{fiq}$$

for some family $\overline{\alpha}$ of parameters, where the right-hand side is a parameterized quantum case statement. In particular, if we further assume that Q contains no measurements, then

$$\mathbf{qif}\,[\overline{q}\,]\,(\square i \cdot |i\rangle \to S_i)\ \mathbf{fiq}; Q = \mathbf{qif}\,[\overline{q}\,]\,(\square i \cdot |i\rangle \to (S_i; Q))\,\mathbf{fiq}.$$

A quantum choice is defined as a “coin” program followed by a quantum case statement. A natural question would be: is it possible to move the “ coin” program to the end of a quantum case statement? The following theorem positively answers this question under the condition that encapsulation in a block with local variables is allowed.

Theorem 6.6.2. *For any programs S_i and unitary operator U, we have:*

$$[U[\overline{q}]]\left(\bigoplus_{i=1}^{n}|i\rangle \to S_i\right) = \mathbf{qif}\ (\square i \cdot U_{\overline{q}}^{\dagger}|i\rangle \to S_i)\ \mathbf{fiq};U[\overline{q}]. \tag{6.28}$$

More generally, for any programs S_i and S with $\overline{q} = qvar(S)$, there are new quantum variables $\overline{r}$, a pure state $|\varphi_0\rangle \in \mathcal{H}_{\overline{r}}$, an orthonormal basis $\{|\psi_{ij}\rangle\}$ of $\mathcal{H}_{\overline{q}} \otimes \mathcal{H}_{\overline{r}}$, programs Q_{ij}, and a unitary operator U in $\mathcal{H}_{\overline{q}} \otimes \mathcal{H}_{\overline{r}}$ such that

$$\begin{aligned}[S]\left(\bigoplus_{i=1}^{n}|i\rangle \to S_i\right) = \mathbf{begin\ local}\ &\overline{r} := |\varphi_0\rangle;\\ &\quad\mathbf{qif}\ \left(\square i,j \cdot |\psi_{ij}\rangle \to Q_{ij}\right)\ \mathbf{fiq};\\ &\quad U[\overline{q},\overline{r}]\\ \mathbf{end}&.\end{aligned} \tag{6.29}$$

The next theorem is the counterpart of Theorem 6.6.1 for quantum choice, showing that quantum choice is also idempotent, commutative and associative, and sequential composition is distributive over quantum choice from the right.

Theorem 6.6.3 (Laws for Quantum Choice).

(i) *Idempotent Law: If $qvar(Q) = \overline{q}$, $tr[\![Q]\!](\rho) = 1$ and $S_i = S$ for all $1 \le i \le n$, then*

$$\mathbf{begin\ local}\ \overline{q} := \rho; [Q]\left(\bigoplus_{i=1}^{n}|i\rangle \to S_i\right)\ \mathbf{end} = S.$$

(ii) *Commutative Law: For any permutation τ of $\{1,\dots,n\}$, we have:*

$$[S]\left(\bigoplus_{i=1}^{n}|i\rangle \to S_{\tau(i)}\right) = [S;U_\tau[\overline{q}]]\left(\bigoplus_{i=1}^{n}|i\rangle \to S_i\right);U_{\tau^{-1}}[\overline{q}],$$

where $qvar(S) = \overline{q}$, and U_τ, $U_{\tau^{-1}}$ are the same as in Theorem 6.6.1 (2).

(iii) *Associative Law: Let*

$$\Gamma = \{(i,j_i) : 1 \le i \le m\ \text{ and } 1 \le j_i \le n_i\} = \bigcup_{i=1}^{m}(\{i\} \times \{1,\dots,n_i\})$$

and

$$R = [S]\left(\bigoplus_{i=1}^{n}|i\rangle \to Q_i\right).$$

Then

$$\left(\bigoplus_{i=1}^{m}|i\rangle \to [Q_i]\left(\bigoplus_{j_i=1}^{n_i}|j_i\rangle \to R_{ij_i}\right)\right) = [R(\overline{\alpha})]\left(\bigoplus_{(i,j_i)\in\Gamma}|i,j_i\rangle \to R_{ij_i}\right),$$

for some family $\overline{\alpha}$ of parameters, where the right-hand side is a parameterized quantum choice defined in Subsection 6.8.1.

(i) *Distributive Law: If $qvar(S) \cap qvar(Q) = \emptyset$, then*

$$[S]\left(\bigoplus_{i=1}^{n}|i\rangle \to S_i\right); Q =_{CF} [S(\overline{\alpha})]\left(\bigoplus_{i=1}^{n}|i\rangle \to (S_i; Q)\right)$$

for some family $\overline{\alpha}$ of parameters, where the right-hand side is a parameterized quantum choice. Here, symbol "$=_{CF}$" stands for "coin-free" equivalence (see Definition 6.4.4). In particular, if we further assume that Q contains no measurements, then

$$[S]\left(\bigoplus_{i=1}^{n}|i\rangle \to S_i\right); Q = [S]\left(\bigoplus_{i=1}^{n}|i\rangle \to (S_i; Q)\right).$$

6.7 ILLUSTRATIVE EXAMPLES

A theory of programming with quantum case statements and quantum choice has been developed in the previous sections, using the quantum programming language QuGCL. In this section, we give some examples to show how some quantum algorithms can be conveniently written as programs in the language QuGCL.

6.7.1 QUANTUM WALKS

The design of the language QuGCL, in particular the definition of quantum case statement and quantum choice, was inspired by the construction of some simplest quantum walks. A large number of variants and generalizations of quantum walks have been introduced in the last decade. Quantum walks have been widely used in the development of quantum algorithms including quantum simulation. Various extended quantum walks in the literature can be easily written as QuGCL programs. Here, we only present several simple examples.

Example 6.7.1. *Recall from Example 2.3.1 that the Hadamard walk is a quantum generalization of a one-dimensional random walk. Let p, c be the quantum variables for position and coin, respectively. The type of variable p is the infinite-dimensional Hilbert space*

$$\mathcal{H}_p = span\{|n\rangle : n \in \mathbb{Z}\ (integers)\} = \left\{\sum_{n=-\infty}^{\infty} \alpha_n|n\rangle : \sum_{n=-\infty}^{\infty} |\alpha_n|^2 < \infty\right\},$$

and the type of c is the 2-dimensional Hilbert space $\mathcal{H}_c = span\{|L\rangle, |R\rangle\}$, where L, R stand for Left and Right, respectively. The state space of the Hadamard walk is $\mathcal{H} = \mathcal{H}_c \otimes \mathcal{H}_p$. Let $I_{\mathcal{H}_p}$ be the identity operator in $\mathcal{H}_p$, H the 2×2 Hadamard matrix and T_L, T_R the left- and right-translations, respectively; that is,

$$T_L|n\rangle = |n-1\rangle, \qquad T_R|n\rangle = |n+1\rangle$$

for every $n \in \mathbb{Z}$. Then a single step of the Hadamard walk can be described by the unitary operator

$$W = (|L\rangle\langle L| \otimes T_L + |R\rangle\langle R| \otimes T_R)(H \otimes I_{\mathcal{H}_p}). \tag{6.30}$$

It can also be written as the QuGCL program:

$$\begin{aligned} T_L[p]_{H[c]} \oplus T_R[p] \equiv H[c];\ \mathbf{qif}\ [c]\ &|L\rangle \rightarrow T_L[p] \\ \square\ \ &|R\rangle \rightarrow T_R[p] \\ \mathbf{fiq}&. \end{aligned}$$

This program is the quantum choice of the left-translation T_L and the right-translation T_R according to the "coin" program $H[c]$. The Hadamard walk repeatedly runs this program.

The following are several variants of this walk considered in the recent physics literature.

(i) *A simple variant of the Hadamard walk is the unidirectional quantum walk, where the walker either moves to the right or stays in the previous position. So, the left-translation T_L should be replaced by the program* **skip** *whose semantics is the identity operator $I_{\mathcal{H}_p}$, and a single step of the new quantum walk can be written as the QuGCL program:*

$$\mathbf{skip}_{H[c]} \oplus T_R[p].$$

It is a quantum choice of **skip** *and the right-translation T_R.*

(ii) *A feature of the Hadamard walk and its unidirectional variant is that the "coin tossing operator" H is independent of the position and time. A new kind of quantum walk was proposed, where the "coin tossing" operator depends on both position n and time t:*

$$C(n,t) = \frac{1}{\sqrt{2}}\begin{pmatrix} c(n,t) & s(n,t) \\ s^*(n,t) & -e^{i\theta}c(n,t) \end{pmatrix}.$$

Then for a given time t, step t of the walk can be written as the QuGCL program:

$$\begin{aligned} W_t \equiv\ & \mathbf{qif}\ [p](\square n \cdot |n\rangle \to C(n,t)[c])\ \mathbf{fiq}; \\ & \mathbf{qif}\ [c]\ |L\rangle \to T_L[p] \\ & \square \qquad |R\rangle \to T_R[p] \\ & \mathbf{fiq}. \end{aligned}$$

The program W_t is a sequential composition of two quantum case statements. In the first quantum case statement, " coin-tossing" program $C(n,t)$ is selected to execute on quantum coin variable c according to position $|n\rangle$, and a superposition of different positions is allowed. Furthermore, since W_t may be different for different time points t, the first T steps can be written as the program:

$$W_1; W_2; \ldots; W_T.$$

(iii) *Another simple generalization of the Hadamard walk is the quantum walk with three "coin" states. The " coin" space of this walk is a 3-dimensional Hilbert space $\mathcal{H}_c = span\{|L\rangle, |0\rangle, |R\rangle\}$, where L and R are used to indicate moving to the left and to the right, respectively, as before, but 0 means staying at the previous position. The "coin tossing" operator is the unitary*

$$U = \frac{1}{3}\begin{pmatrix} -1 & 2 & 2 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{pmatrix}.$$

Then a single step of the walk can be written as the QuGCL program:

$$[U[c]]\ (|L\rangle \to T_L[p] \oplus |0\rangle \to \mathbf{skip} \oplus |R\rangle \to T_R[p])\,.$$

This is the quantum choice of **skip***, the left- and right-translations according to the "coin" program $U[c]$.*

The quantum walks in this example have only a single walker as well as a single "coin." In the following two examples, we consider some more complicated quantum walks in which multiple walkers participate and multiple "coins" are equipped to control the walkers.

Example 6.7.2. *We consider a one-dimensional quantum walk driven by multiple "coins." In this walk, there is still a single walker, but it is controlled by M different "coins". Each of these "coins" has its own state space, but the "coin tossing" operator for all of them is the same, namely the 2×2 Hadamard matrix. Now let variable p, Hilbert spaces $\mathcal{H}_p, \mathcal{H}_c$ and operators T_L, T_R, H be the same as in Example 6.7.1, and let $c_1, \ldots, c_M$ be the quantum variables for the M "coins." Then the state space of the walk is*

$$\mathcal{H} = \bigotimes_{m=1}^{M} \mathcal{H}_{c_m} \otimes \mathcal{H}_p,$$

where $\mathcal{H}_{c_m} = \mathcal{H}_c$ for all $1 \le m \le M$. We write

$$W_m \equiv \left(T_L[p]_{H[c_1]} \oplus T_R[p]\right); \ldots ; \left(T_L[p]_{H[c_m]} \oplus T_R[p]\right)$$

for $1 \le m \le M$. If we cycle among the M "coins," starting from the "coin" c_1, then the first T steps of the walk can be written in the language QuGCL as follows:

$$W_M; \ldots ; W_M; W_r$$

where W_M is iterated for $d = \lfloor T/M \rfloor$ times, and $r = T - Md$ is the remainder of T divided by M. This program is a sequential composition of T quantum choices of the left- and right-translations controlled by different "coins."

Example 6.7.3. *We consider a quantum walk consisting of two walkers on a line sharing "coins." The two walkers have different state spaces, and each of them has its own "coin." So, the state Hilbert space of the whole quantum walk is $\mathcal{H}_c \otimes \mathcal{H}_c \otimes \mathcal{H}_p \otimes \mathcal{H}_p$. If the two walkers are completely independent, then the step operator of this walk is $W \otimes W$, where W is defined by equation (6.30). But more interesting is the case where a two-qubit unitary operator U is introduced to entangle the two "coins." This case can be thought of as that the two walkers are sharing "coins." A step of this quantum walk can be written as a QuGCL program as follows:*

$$U[c_1, c_2]; \left(T_L[q_1]_{H[c_1]} \oplus T_R[q_1]\right); \left(T_L[q_2]_{H[c_2]} \oplus T_R[q_2]\right)$$

where q_1, q_2 are the position variables and c_1, c_2 the "coin" variables of the two walkers, respectively. Here, the two walkers both use the Hadamard operator H for " coin-tossing."

Obviously, a generalization to the case with more than two walkers can also be easily programmed in QuGCL.

6.7.2 QUANTUM PHASE ESTIMATION

Not only quantum walk-based algorithms can be conveniently programmed in the language QuGCL. In this subsection, we show how to program a quantum phase estimation algorithm in QuGCL. Recall the algorithm from Subsection 2.3.7, given a unitary operator U and its eigenvector $|u\rangle$. The goal of this algorithm is to estimate the phase φ of the eigenvalue $e^{2\pi i\varphi}$ corresponding to $|u\rangle$. The algorithm is described in Figure 6.1. Here, for each $j = 0, 1, \ldots, t-1$, an oracle performs the controlled-U^{2^j} operator, and $FT^\dagger$ stand for the inverse quantum Fourier transform.

- **Procedure:**

1. $|0\rangle^{\otimes t}|u\rangle \xrightarrow{H^{\otimes} \text{ on the first } t \text{ qubits}} \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|u\rangle$
2. $\xrightarrow{\text{oracles}} \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle U^j|u\rangle = \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}e^{2\pi ij\varphi}|j\rangle|u\rangle$
3. $\xrightarrow{FT^{\dagger}} \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}e^{2\pi ij\varphi}\left(\frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}e^{-2\pi ijk/2^t}|k\rangle\right)|u\rangle$
4. $\xrightarrow{\text{measure the first } t \text{ qubits}} |m\rangle|u\rangle,$

FIGURE 6.1

Quantum phase estimation.

Now we use qubit variables $q_1, \dots, q_t$ as well as a quantum variable p of which the type is the Hilbert space of unitary operator U. We also use a classical variable to record the outcomes of a measurement. Then quantum phase estimation can be written as the QuGCL program in Figure 6.2, where:

- **Program:**

1. $\mathbf{skip}_{H[c_1]} \oplus S_1;$

   $\dots\dots$

2. $\mathbf{skip}_{H[c_t]} \oplus S_t;$
3. $q_t := H[q_t];$
4. $T_t;$
5. $q_{t-1} := H[q_{t-1}];$
6. $T_{t-1};$
7. $q_{t-2} := H[q_{t-2}];$

   $\dots\dots$

8. $T_2;$
9. $q_1 := H[q_1];$
10. $\mathbf{if}\ (\square\ M[q_1, \dots, q_t : x] = m \rightarrow \mathbf{skip})\ \mathbf{fi}$

FIGURE 6.2

Quantum phase estimation program.

- for $1 \leq k \leq t$,

$$S_k \equiv q := U[q]; \ldots ; q := U[q]$$

(the sequential composition of 2^{k-1} copies of unitary transformation U);
- for $2 \leq k \leq t$, the subprogram T_k is given in Figure 6.3, and the operator $R_k^\dagger$ in T_k is given by

$$R_k^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & e^{-2\pi i/2^k} \end{pmatrix};$$

```
1.  qif [$q_t$] $|0\rangle \rightarrow$ skip
2.  □          $|1\rangle \rightarrow R_k^\dagger[q_{k-1}]$
3.  fiq;
4.  qif [$q_{t-1}$] $|0\rangle \rightarrow$ skip
5.  □            $|1\rangle \rightarrow R_{k-1}^\dagger[q_{k-1}]$
6.  fiq;
      ......
7.  qif [$q_{k+1}$] $|0\rangle \rightarrow$ skip
9.  □            $|1\rangle \rightarrow R_2^\dagger[q_{k-1}]$
10. fiq
```

FIGURE 6.3

Subprogram T_k.

- $M = \{M_m : m \in \{0,1\}^t\}$ is the measurement on t qubits in the computational basis; that is, $M_m = |m\rangle\langle m|$ for every $m \in \{0,1\}^t$.
- Subprograms $T_2, \ldots T_k$ are displayed in Figure 6.3.

It can be seen from equation (2.24) that the part made up of lines 3-9 in the phase estimation program (Figure 6.2) is actually the inverse quantum Fourier transform. Since the language QuGCL does not include any initialization statement, $|0\rangle^t|u\rangle$ in Figure 6.1 can only be seen as an input to the program.

6.8 DISCUSSIONS

In the previous sections, the two program constructs of quantum case statement and quantum choice have been thoroughly studied, and they were used to program quantum walk-based algorithms and quantum phase estimation. It was mentioned in

Subsection 6.3.3 that a choice of coefficients different from that in Definition 6.3.5 is possible, which implies the possibility of a different semantics of a quantum case statement. This section is devoted to discussing several variants of quantum case statement and quantum choice. These variants can only appear in the quantum setting and have no counterparts in classical programming. They are both conceptually interesting and useful in applications. Indeed, some of these variants were already used in the statements of several algebraic laws in Section 6.6.

6.8.1 COEFFICIENTS IN GUARDED COMPOSITIONS OF QUANTUM OPERATIONS

The coefficients in the right-hand side of the defining equation (6.14) of guarded composition of operator-valued functions are chosen in a very special way, with a physical interpretation in terms of conditional probability. This subsection shows that other choices of these coefficients are possible.

Let us first consider the simplest case: the guarded composition

$$U \triangleq \bigoplus_{k=1}^{n}(|k\rangle \rightarrow U_k)$$

of unitary operators U_k $(1 \leq k \leq n)$ in a Hilbert space $\mathcal{H}$ along an orthonormal basis $\{|k\rangle\}$ of a "coin" Hilbert space $\mathcal{H}_c$. If for each $1 \leq k \leq n$, we add a relative phase θ_k into the defining equation (6.10) of U:

$$U(|k\rangle|\psi\rangle) = e^{i\theta_k}|k\rangle U_k|\psi\rangle \tag{6.31}$$

for all $|\psi\rangle \in \mathcal{H}$, then equation (6.11) is changed to

$$U\left(\sum_k \alpha_k|k\rangle|\psi_k\rangle\right) = \sum_k \alpha_k e^{i\theta_k}|k\rangle U_k|\psi_k\rangle. \tag{6.32}$$

Note that phases θ_k in equation (6.31) can be different for different basis states $|k\rangle$. It is easy to see that the new operator U defined by equation (6.31) or (6.32) is still unitary.

The idea of adding relative phases also applies to the guarded composition of operator-valued functions. Consider

$$F \triangleq \bigoplus_{k=1}^{n}(|k\rangle \rightarrow F_k)$$

where $\{|k\rangle\}$ is an orthonormal basis of $\mathcal{H}_c$, and F_k is an operator-valued function in $\mathcal{H}$ over Δ_k for every $1 \leq k \leq n$. We arbitrarily choose a sequence $\theta_1, \ldots, \theta_n$ of real numbers and change the defining equation (6.14) of F to

$$F(\oplus_{k=1}^{n}\delta_k)|\Psi\rangle = \sum_{k=1}^{n} e^{i\theta_k}\left(\prod_{l\neq k}\lambda_{l\delta_l}\right)|k\rangle(F_k(\delta_k)|\psi_k\rangle) \tag{6.33}$$

for any state

$$|\Psi\rangle = \sum_{k=1}^{n}|k\rangle|\psi_k\rangle \in \mathcal{H}_c \otimes \mathcal{H},$$

where $\lambda_{l\delta_l}$'s are the same as in Definition 6.3.5. Then it is clear that F defined by equation (6.33) is still an operator-valued function. Indeed, this conclusion is true for a much more general definition of guarded composition of operator-valued functions. Let F_k be an operator-valued function in $\mathcal{H}$ over Δ_k for each $1 \le k \le n$, and let

$$\overline{\alpha} = \left\{\alpha^{(k)}_{\delta_1,\ldots,\delta_{k-1},\delta_{k+1},\ldots,\delta_n} : 1 \le k \le n,\ \delta_l \in \Delta_l\ (l = 1,\ldots,k-1,k+1,\ldots,n)\right\} \tag{6.34}$$

be a family of complex numbers satisfying the normalization condition:

$$\sum_{\delta_1\in\Delta_1,\ldots,\delta_{k-1}\in\Delta_{k-1},\delta_{k+1}\in\Delta_{k+1},\ldots,\delta_n\in\Delta_n}\left|\alpha^{(k)}_{\delta_1,\ldots,\delta_{k-1},\delta_{k+1},\ldots,\delta_n}\right|^2 = 1 \tag{6.35}$$

for every $1 \le k \le n$. Then we can define the $\overline{\alpha}$-guarded composition

$$F \triangleq (\overline{\alpha})\bigoplus_{k=1}^{n}(|i\rangle \to F_k)$$

of F_k $(1 \le k \le n)$ along an orthonormal basis $\{|k\rangle\}$ of $\mathcal{H}_c$ by

$$F\left(\oplus_{k=1}^{n}\delta_k\right)\left(\sum_{k=1}^{n}|k\rangle|\psi_k\rangle\right) = \sum_{k=1}^{n}\alpha^{(k)}_{\delta_1,\ldots,\delta_{k-1},\delta_{k+1},\ldots,\delta_n}|k\rangle\left(F_k(\delta_k)|\psi_k\rangle\right) \tag{6.36}$$

for any $|\psi_1\rangle,\ldots,|\psi_n\rangle \in \mathcal{H}$ and for any $\delta_k \in \Delta_k$ $(1 \le k \le n)$. Note that coefficient

$$\alpha^{(k)}_{\delta_1,\ldots,\delta_{k-1},\delta_{k+1},\ldots,\delta_n}$$

does not contain parameter δ_k. This independence together with condition (6.35) guarantees that the $\overline{\alpha}$-guarded composition is an operator-valued function, as can be seen from the proof of Lemma 6.3.2 presented in Section 6.9.

Example 6.8.1

(i) *Definition 6.3.5 is a special case of $\overline{\alpha}$-guarded composition because if for any $1 \le i \le n$ and $\delta_k \in \Delta_k$ $(k = 1,\ldots,i-1,i+1,\ldots,n)$, we set*

$$\alpha^{i}_{\delta_1,\ldots,\delta_{i-1},\delta_{i+1},\ldots,\delta_n} = \prod_{k\neq i}\lambda_{k\delta_k},$$

where $\lambda_{k\delta_k}$'s are given by equation (6.15), then equation (6.36) degenerates to (6.14).

(ii) *Another possible choice of $\overline{\alpha}$ is*

$$\alpha^{i}_{\delta_1,\dots,\delta_{i-1},\delta_{i+1},\dots,\delta_n} = \frac{1}{\sqrt{\prod_{k\neq i}|\Delta_k|}}$$

for all $1 \le i \le n$ and $\delta_k \in \Delta_k$ $(k = 1,\dots,i-1,i+1,\dots,n)$. Obviously, for this family $\overline{\alpha}$ of coefficients, the $\overline{\alpha}$-guarded composition cannot be obtained by modifying Definition 6.3.5 with relative phases.

Now we are able to define parameterized quantum case statement and quantum choice.

Definition 6.8.1

(i) *Let $\overline{q}$, $\{|i\rangle\}$ and $\{S_i\}$ be as in Definition 6.2.1 (iv). Furthermore, let the classical states $\Delta(S_i) = \Delta_i$ for every i, and let $\overline{\alpha}$ be a family of parameters satisfying condition (6.35), as in equation (6.34). Then the $\overline{\alpha}$-quantum case statement of $S_1,\dots,S_n$ guarded by basis states $|i\rangle$'s is*

$$S \equiv \ \mathbf{qif}\ (\overline{\alpha})[\,\overline{q}\,]\,(\square i\cdot\ |i\rangle \rightarrow S_i)\ \ \mathbf{fiq} \tag{6.37}$$

and its semi-classical semantics is

$$\llbracket S\rrbracket = (\overline{\alpha})\bigoplus_{i=1}^{n}\left(|i\rangle \rightarrow \llbracket S_i\rrbracket\right).$$

(ii) *Let S, $\{|i\rangle\}$ and S_i's be as in Definition 6.5.1, and let $\overline{\alpha}$ be as above. Then the $\overline{\alpha}$-quantum choice of S_i's according to S along the basis $\{|i\rangle\}$ is defined as*

$$[S(\overline{\alpha})]\left(\bigoplus_i |i\rangle \rightarrow S_i\right) \equiv \ S;\mathbf{qif}\ (\overline{\alpha})[\,\overline{q}\,]\,(\square i\cdot|i\rangle \rightarrow S_i)\ \ \mathbf{fiq}.$$

The symbol $[\,\overline{q}\,]$ in quantum case statement (6.37) can be dropped whenever quantum variables $\overline{q}$ can be recognized from the context. At the first glance, it seems unreasonable that the parameters $\overline{\alpha}$ in the syntax (6.37) of $\overline{\alpha}$-quantum case statement are indexed by the classical states of S_i. But this is not problematic at all because the classical states of S_i are completely determined by the syntax of S_i.

The purely quantum denotational semantics of the $\overline{\alpha}$-quantum case statement can be obtained from its semi-classical semantics according to Definition 6.4.3, and the semantics of $\overline{\alpha}$-quantum choice can be derived from the semantics of the $\overline{\alpha}$-quantum case statement. The notions of parameterized quantum case statement and quantum choice were already used in the presentation of several theorems in Section 6.6.

Problem 6.8.1. *Prove or disprove the following statement: for any $\overline{\alpha}$, there exists a unitary operator U such that*

$$\textbf{qif}\ (\overline{\alpha})[\overline{q}]\,(\Box i\cdot\ |i\rangle \rightarrow S_i)\ \textbf{fiq} = [U[\overline{q}]]\left(\bigoplus_i |i\rangle \rightarrow S_i\right).$$

What happens when $U[\overline{q}]$ is replaced by a general quantum program (of which the semantics can be a general quantum operation rather than a unitary)?

6.8.2 QUANTUM CASE STATEMENTS GUARDED BY SUBSPACES

A major difference between case statement (6.2) of classical programs and quantum case statement (6.9) can be revealed by a comparison between their guards: the guards G_i in the former are propositions about the program variables, whereas the guards $|i\rangle$ in the latter are basis states of the "coin" space $\mathcal{H}_c$. However, this difference is not as big as we imagine at first glance. In the Birkhoff-von Neumann quantum logic [42], a proposition about a quantum system is expressed by a closed subspace of the state Hilbert space of the system. This observation leads us to a way to define quantum case statement guarded by propositions about the "coin" system instead of basis states of the " coin" space.

Definition 6.8.2. *Let $\overline{q}$ be a sequence of quantum variables and $\{S_i\}$ be a family of quantum programs such that*

$$\overline{q}\cap\left(\bigcup_i qvar(S_i)\right)=\emptyset.$$

Suppose that $\{X_i\}$ is a family of propositions about the "coin" system $\overline{q}$, i.e., closed subspaces of the " coin" space $\mathcal{H}_{\overline{q}}$, satisfying the following two conditions:

(i) *X_i's are pairwise orthogonal, i.e., $X_{i_1}\perp X_{i_2}$ provided $i_1\neq i_2$;*

(ii) $\bigoplus_i X_i \triangleq span\left(\bigcup_i X_i\right)=\mathcal{H}_{\overline{q}}$.

Then

(i) *The quantum case statement of S_i's guarded by subspaces X_i's:*

$$S\equiv\ \textbf{qif}\ [\overline{q}]\,(\Box i\cdot X_i\rightarrow S_i)\ \textbf{fiq} \tag{6.38}$$

is a program.

(ii) *The quantum variables of S are:*

$$qvar(S)=\overline{q}\cup\left(\bigcup_i qvar(S_i)\right).$$

(iii) *The purely quantum denotational semantics of the quantum case statement is:*

$$[\![S]\!] = \{[\![\mathbf{qif}\ [\,\overline{q}\,]\ \big(\Box i, j_i \cdot |\varphi_{ij_i}\rangle \to S_{ij_i}\big)\ \mathbf{fiq}]\!] : \{|\varphi_{ij_i}\rangle\}\ \textit{ is an orthonormal basis of } X_i \textit{ for each } i, \textit{ and } S_{ij_i} = S_i \textit{ for every } i, j_i\}. \tag{6.39}$$

Intuitively, $\{X_i\}$ in quantum case statement (6.38) can be thought of as a partition of the whole state Hilbert space $\mathcal{H}_{\overline{q}}$. For simplicity, the variables $\overline{q}$ in equation (6.38) can be dropped if they can be recognized from the context. It is clear that the (disjoint) union $\bigcup_i\{|\varphi_{ij_i}\rangle\}$ of the bases of subspaces X_i's in equation (6.39) is an orthonormal basis of the whole " coin" space $\mathcal{H}_c$. From the right-hand side of equation (6.39), we note that the purely quantum semantics of program (6.38) guarded by subspaces is a set of quantum operations rather than a single quantum operation. So, quantum case statement (6.38) is a nondeterministic program, and its nondeterminism comes from different choices of the bases of guard subspaces. Furthermore, a quantum case statement guarded by basis states of these subspaces is a refinement of quantum case statement (6.38). On the other hand, if $\{|i\rangle\}$ is an orthonormal basis of $\mathcal{H}_{\overline{q}}$, and for each i, X_i is the one-dimensional subspace $span\{|i\rangle\}$, then the previous definition degenerates to the quantum case statement (6.9) guarded by basis states $|i\rangle$.

The notion of equivalence for quantum programs in Definition 6.4.4 can be easily generalized to the case of nondeterministic quantum programs (i.e., programs with a set of quantum operations rather than a single quantum operation as its semantics) provided we make the following conventions:

- If Ω is a set of quantum operations and $\mathcal{F}$ a quantum operation, then
$$\Omega \otimes \mathcal{F} = \{\mathcal{E} \otimes \mathcal{F} : \mathcal{E} \in \Omega\};$$
- We identify a single quantum operation with the set containing only this quantum operation.

Some basic properties of quantum case statement guarded by subspaces are given in the following:

Proposition 6.8.1

(i) *If for every i, S_i does not contain any measurement, then for any orthonormal basis $\{|\varphi_{ij_i}\rangle\}$ of X_i $(1 \leq i \leq n)$, we have:*

$$\mathbf{qif}\ (\Box i \cdot X_i \to S_i)\ \mathbf{fiq} = \mathbf{qif}\ \big(\Box i, j_i \cdot |\varphi_{ij_i}\rangle \to S_{ij_i}\big)\ \mathbf{fiq}$$

where $S_{ij_i} = S_i$ for every i, j_i. In particular, if for every i, U_i is a unitary operator in $\mathcal{H}_{\overline{q}}$, then

$$\mathbf{qif}\ [\overline{p}](\Box i \cdot X_i \to U_i[\,\overline{q}\,])\ \mathbf{fiq} = U[\overline{p}, \overline{q}]$$

where

$$U = \sum_i (I_{X_i} \otimes U_i)$$

is an unitary operator in $\mathcal{H}_{\overline{p} \cup \overline{q}}$.

(ii) *Let U be a unitary operator in $\mathcal{H}_{\overline{q}}$. If for every i, X_i is an invariant subspace of U, i.e.,*

$$UX_i = \{U|\psi\rangle : |\psi\rangle \in X_i\} \subseteq X_i,$$

then

$$U[\overline{q}]; \textbf{qif}\,[\overline{q}]\,(\Box i \cdot X_i \rightarrow S_i)\ \textbf{fiq}; U^\dagger[\overline{q}] = \textbf{qif}\,[\overline{q}]\,(\Box i \cdot X_i \rightarrow S_i)\ \textbf{fiq}.$$

Exercise 6.8.1. *Prove Proposition 6.8.1.*

We conclude this section by pointing out that a generalized notion of quantum choice can be defined based on either a parameterized quantum case statement or quantum case statement guarded by subspaces, and the algebraic laws established in Section 6.6 can be easily generalized for a parameterized quantum case statement and quantum choice as well as those guarded by subspaces. The details are omitted here, but the reader can try to figure it out as an exercise.

6.9 PROOFS OF LEMMAS, PROPOSITIONS AND THEOREMS

The proofs of the main lemmas, propositions and theorems in this chapter all include tedious calculations. So, for readability, these results have been stated in the previous sections without proofs. To complete this chapter, here we present their detailed proofs.

Proof of Lemma 6.3.2. Let F be the operator-valued function given in Definition 6.3.5. We write:

$$\overline{F} \triangleq \sum_{\delta_1\in\Delta_1,\ldots,\delta_n\in\Delta_n} F(\oplus_{i=1}^n \delta_i)^\dagger \cdot F(\oplus_{i=1}^n \delta_i).$$

Let us start with an auxiliary equality. For any $|\Phi\rangle, |\Psi\rangle \in \mathcal{H}_c \otimes \mathcal{H}$, we can write

$$|\Phi\rangle = \sum_{i=1}^n |i\rangle|\varphi_i\rangle \ \text{ and } |\Psi\rangle = \sum_{i=1}^n |i\rangle|\psi_i\rangle$$

where $|\varphi_i\rangle, |\psi_i\rangle \in \mathcal{H}$ for each $1 \leq i \leq n$. Then we have:

$$\begin{aligned}\langle\Phi|\overline{F}|\Psi\rangle &= \sum_{\delta_1,\dots,\delta_n}\langle\Phi|F(\oplus_{i=1}^n\delta_i)^\dagger\cdot F(\oplus_{i=1}^n\delta_i)|\Psi\rangle\\ &= \sum_{\delta_1,\dots,\delta_n}\sum_{i,i'=1}^n\left(\prod_{k\neq i}\lambda_{k\delta_k}^*\right)\left(\prod_{k\neq i'}\lambda_{k\delta_k}\right)\langle i|i'\rangle\langle\varphi_i|F_i(\delta_i)^\dagger F_{i'}(\delta_{i'})|\psi_{i'}\rangle\\ &= \sum_{\delta_1,\dots,\delta_n}\sum_{i=1}^n\left(\prod_{k\neq i}|\lambda_{k\delta_k}|^2\right)\langle\varphi_i|F_i(\delta_i)^\dagger F_i(\delta_i)|\psi_i\rangle\\ &= \sum_{i=1}^n\left[\sum_{\delta_1,\dots,\delta_{i-1},\delta_{i+1},\dots,\delta_n}\left(\prod_{k\neq i}|\lambda_{k\delta_k}|^2\right)\cdot\sum_{\delta_i}\langle\varphi_i|F_i(\delta_i)^\dagger F_i(\delta_i)|\psi_i\rangle\right]\\ &= \sum_{i=1}^n\sum_{\delta_i}\langle\varphi_i|F_i(\delta_i)^\dagger F_i(\delta_i)|\psi_i\rangle\\ &= \sum_{i=1}^n\langle\varphi_i|\sum_{\delta_i}F_i(\delta_i)^\dagger F_i(\delta_i)|\psi_i\rangle\end{aligned}\tag{6.40}$$

because for each k, we have:

$$\sum_{\delta_k}|\lambda_{k\delta_k}|^2 = 1,$$

and thus

$$\sum_{\delta_1,\dots,\delta_{i-1},\delta_{i+1},\dots,\delta_n}\left(\prod_{k\neq i}|\lambda_{k\delta_k}|^2\right) = \prod_{k\neq i}\left(\sum_{\delta_k}|\lambda_{k\delta_k}|^2\right) = 1.\tag{6.41}$$

Now we are ready to prove our conclusions by using equation (6.40).

(i) We first prove that $\overline{F}\sqsubseteq I_{\mathcal{H}_c\otimes\mathcal{H}}$, i.e., F is an operator-valued function in $\mathcal{H}_c\otimes\mathcal{H}$ over $\bigoplus_{i=1}^n\Delta_n$. It suffices to show that

$$\langle\Phi|\overline{F}|\Phi\rangle\leq\langle\Phi|\Phi\rangle$$

for each $|\Phi\rangle\in\mathcal{H}_c\otimes\mathcal{H}$. In fact, for each $1\leq i\leq n$, since F_i is an operator-valued function, we have:

$$\sum_{\delta_i}F_i(\delta_i)^\dagger F_i(\delta_i)\sqsubseteq I_{\mathcal{H}}.$$

Therefore, it holds that

$$\langle\varphi_i|\sum_{\delta_i}F_i(\delta_i)^\dagger F_i(\delta_i)|\varphi_i\rangle\leq\langle\varphi_i|\varphi_i\rangle.$$

Then it follows immediately from equation (6.40) that

$$\langle\Phi|\overline{F}|\Phi\rangle \leq \sum_{i=1}^{n}\langle\varphi_i|\varphi_i\rangle = \langle\Phi|\Phi\rangle.$$

So, F is an operator-valued function.

(ii) Secondly, we prove that F is full for the case where all F_i $(1 \leq i \leq n)$ are full. It requires us to show that $\overline{F} = I_{\mathcal{H}_c\otimes\mathcal{H}}$. In fact, for every $1 \leq i \leq n$, we have:

$$\sum_{\delta_i} F_i(\delta_i)^\dagger F_i(\delta_i) = I_{\mathcal{H}}$$

because F_i is full. Thus, it follows from equation (6.40) that

$$\langle\Phi|\overline{F}|\Psi\rangle = \sum_{i=1}^{n}\langle\varphi_i|\psi_i\rangle = \langle\Phi|\Psi\rangle$$

for any $|\Phi\rangle, |\Psi\rangle \in \mathcal{H}_c \otimes \mathcal{H}$. So, it holds that $\overline{F} = I_{\mathcal{H}_c\otimes\mathcal{H}}$ by arbitrariness of $|\Phi\rangle$ and $\Psi\rangle$, and F is full.

Proof of *Proposition 6.4.1*. Clauses (i) to (iv) are obvious, and we only prove (v) and (vi).

(1) To prove clause (v), let

$$S \equiv \textbf{if } (\Box m \cdot M[\overline{q} : x] = m \rightarrow S_m) \textbf{ fi}.$$

Then by Definitions 6.4.2 and 6.4.3, for any partial density operator ρ in $\mathcal{H}_{qvar(S)}$, we have:

$$\begin{aligned}
[\![S]\!](\rho) &= \sum_m \sum_{\delta\in\Delta(S_m)} [\![S]\!](\delta[x \leftarrow m])\rho[\![S]\!](\delta[x \leftarrow m])^\dagger \\
&= \sum_m \sum_{\delta\in\Delta(S_m)} \left([\![S_m]\!](\delta) \otimes I_{qvar(S)\setminus qvar(S_m)}\right)\left(M_m \otimes I_{qvar(S)\setminus\overline{q}}\right) \\
&\qquad \rho\left(M_m^\dagger \otimes I_{qvar(S)\setminus\overline{q}}\right)\left([\![S_m]\!](\delta)^\dagger \otimes I_{qvar(S)\setminus qvar(S_m)}\right) \\
&= \sum_m \sum_{\delta\in\Delta(S_m)} \left([\![S_m]\!](\delta) \otimes I_{qvar(S)\setminus qvar(S_m)}\right)\left(M_m\rho M_m^\dagger\right) \\
&\qquad \left([\![S_m]\!](\delta)^\dagger \otimes I_{qvar(S)\setminus qvar(S_m)}\right) \\
&= \sum_m [\![S_m]\!]\left(M_m\rho M_m^\dagger\right) \\
&= \left(\sum_m \left[(M_m \circ M_m^\dagger); [\![S_m]\!]\right]\right)(\rho).
\end{aligned}$$

(2) Finally, we prove clause (vi). For simplicity of the presentation, we write:

$$S \equiv \textbf{qif } [\overline{q}](\Box i \cdot |i\rangle \rightarrow S_i) \textbf{ fiq}.$$

By Definitions 6.4.2, we obtain:

$$\llparenthesis S \rrparenthesis = \bigoplus_i \left(|i\rangle \rightarrow \llparenthesis S_i \rrparenthesis \right).$$

Note that $\llparenthesis S_i \rrparenthesis \in \mathbb{F}(\llbracket S_i \rrbracket)$ for every $1 \le i \le n$, where $\mathbb{F}(\mathcal{E})$ stands for the set of operator-valued functions generated by quantum operation $\mathcal{E}$ (see Definition 6.3.3). Therefore, it follows from Definition 6.4.3 that

$$\begin{aligned}\llbracket S \rrbracket = \mathcal{E}(\llparenthesis S \rrparenthesis) &\in \left\{ \mathcal{E}\left(\bigoplus_i (|i\rangle \rightarrow F_i) \right) : F_i \in \mathbb{F}(\llbracket S_i \rrbracket) \ \text{ for every } i \right\} \\ &= \bigoplus_i \left(|i\rangle \rightarrow \llbracket S_i \rrbracket \right).\end{aligned}$$

Proof of Theorem 6.5.1. To simplify the presentation, we write:

$$R \equiv \ \textbf{qif}\ [\overline{q}](\square i \cdot |i\rangle \rightarrow S_i)\ \textbf{fiq}.$$

To prove the equivalence of two QuGCL programs, we have to show that their purely quantum semantics are the same. However, purely quantum semantics is defined in terms of semi-classical semantics (see Definition 6.4.3). So, we need to work at the level of semi-classical semantics first, and then lift it to the purely quantum semantics. Assume that the semi-classical semantics $\llparenthesis S_i \rrparenthesis$ is the operator-valued function over Δ_i such that

$$\llparenthesis S_i \rrparenthesis(\delta_i) = E_{i\delta_i}$$

for each $\delta_i \in \Delta_i$. Let states $|\psi\rangle \in \mathcal{H}_{\bigcup_{i=1}^n qvar(S_i)}$ and $|\varphi\rangle \in \mathcal{H}_{\overline{q}}$. We can write $|\varphi\rangle = \sum_{i=1}^n \alpha_i |i\rangle$ for some complex numbers α_i $(1 \le i \le n)$. Then for any $\delta_i \in \Delta_i$ $(1 \le i \le n)$, we have:

$$\begin{aligned}|\Psi_{\delta_1\ldots\delta_n}\rangle &\stackrel{\triangle}{=} \llparenthesis R \rrparenthesis(\oplus_{i=1}^n \delta_i)(|\varphi\rangle|\psi\rangle) \\ &= \llparenthesis R \rrparenthesis(\oplus_{i=1}^n \delta_i)\left(\sum_{i=1}^n \alpha_i |i\rangle|\psi\rangle \right) \\ &= \sum_{i=1}^n \alpha_i \left(\prod_{k\neq i} \lambda_{k\delta_k} \right) |i\rangle(E_{i\sigma_i}|\psi\rangle)\end{aligned}$$

where $\lambda_{i\delta_i}$'s are defined as in equation (6.15). We continue to compute:

$$|\Psi_{\delta_1\ldots\delta_n}\rangle\langle\Psi_{\delta_1\ldots\delta_n}| = \sum_{i,j=1}^n \left[\alpha_i\alpha_j^* \left(\prod_{k\neq i} \lambda_{k\delta_k} \right) \left(\prod_{k\neq j} \lambda_{k\delta_k} \right) |i\rangle\langle j| \otimes E_{i\delta_i}|\psi\rangle\langle\psi|E_{j\delta_j}^\dagger \right],$$

and it follows that

$$tr_{\mathcal{H}_{\overline{q}}}|\Psi_{\delta_1\dots\delta_n}\rangle\langle\Psi_{\delta_1\dots\delta_n}| = \sum_{i=1}^{n}|\alpha_i|^2\left(\prod_{k\neq i}\lambda_{k\delta_k}\right)^2 E_{i\delta_i}|\psi\rangle\langle\psi|E_{i\delta_i}^{\dagger}.$$

Using equation (6.41), we obtain:

$$\begin{aligned}
tr_{\mathcal{H}_{\overline{q}}}[\![R]\!](|\varphi\psi\rangle\langle\varphi\psi|) &= tr_{\mathcal{H}_{\overline{q}}}\left(\sum_{\delta_1,\dots,\delta_n}|\Psi_{\delta_1\dots\delta_n}\rangle\langle\Psi_{\delta_1\dots\delta_n}|\right)\\
&= \sum_{\delta_1,\dots,\delta_n} tr_{\mathcal{H}_{\overline{q}}}|\Psi_{\delta_1\dots\delta_n}\rangle\langle\Psi_{\delta_1\dots\delta_n}|\\
&= \sum_{i=1}^{n}|\alpha_i|^2\left[\sum_{\delta_1,\dots,\delta_{i-1},\delta_{i+1},\dots,\delta_n}\left(\prod_{k\neq i}\lambda_{k\delta_k}\right)^2\right]\cdot\left[\sum_{\delta_i}E_{i\delta_i}|\psi\rangle\langle\psi|E_{i\delta_i}^{\dagger}\right]\\
&= \sum_{i=1}^{n}|\alpha_i|^2[\![S_i]\!](|\psi\rangle\langle\psi|).
\end{aligned} \tag{6.42}$$

Now we do spectral decomposition for $[\![S]\!](\rho)$, which is a density operator, and assume that

$$[\![S]\!](\rho) = \sum_l s_l|\varphi_l\rangle\langle\varphi_l|.$$

We further write $|\varphi_l\rangle = \sum_i \alpha_{li}|i\rangle$ for every l. For any density operator σ in $\mathcal{H}_{\bigcup_{i=1}^{n} qvar(S_i)}$, we can write σ in the form of $\sigma = \sum_m r_m|\psi_m\rangle\langle\psi_m|$. Then using equation (6.42), we get:

$$\begin{aligned}
&[\![\mathbf{begin\ local}\ \overline{q} := \rho; [S]\left(\bigoplus_{i=1}^{n}|i\rangle\rightarrow S_i\right)\ \mathbf{end}]\!](\sigma)\\
&= tr_{\mathcal{H}_{\overline{q}}}[\![S;R]\!](\sigma\otimes\rho) = tr_{\mathcal{H}_{\overline{q}}}[\![R]\!](\sigma\otimes[\![S]\!](\rho))\\
&= tr_{\mathcal{H}_{\overline{q}}}[\![R]\!]\left(\sum_{m,l}r_m s_l|\psi_m\varphi_l\rangle\langle\psi_m\varphi_l|\right)\\
&= \sum_{m,l}r_m s_l tr_{\mathcal{H}_{\overline{q}}}[\![R]\!](|\psi_m\varphi_l\rangle\langle\psi_m\varphi_l|)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{m,l} r_m s_l \sum_{i=1}^{n} |\alpha_{li}|^2 [\![S_i]\!](|\psi_m\rangle\langle\psi_m|) \\
&= \sum_{l}\sum_{i=1}^{n} s_l|\alpha_{li}|^2 [\![S_i]\!]\left(\sum_m r_m|\psi_m\rangle\langle\psi_m|\right) \\
&= \sum_{l}\sum_{i=1}^{n} s_l|\alpha_{li}|^2 [\![S_i]\!](\sigma) \\
&= \sum_{i=1}^{n}\left(\sum_l s_l|\alpha_{li}|^2\right)[\![S_i]\!](\sigma) = \left[\!\!\left[\sum_{i=1}^{n} S_i@p_i\right]\!\!\right](\sigma),
\end{aligned}$$

where:

$$\begin{aligned}
p_i = \sum_l s_l|\alpha_{li}|^2 &= \sum_l s_l\langle i|\varphi_l\rangle\langle\varphi_l|i\rangle \\
&= \langle i|\left(\sum_l s_l|\varphi_l\rangle\langle\varphi_l|\right)|i\rangle = \langle i|[\![S]\!](\rho)|i\rangle.
\end{aligned}$$

Proof of Theorem 6.6.2. We first prove the easier part, namely equation (6.28). Let *LHS* and *RHS* stand for the left- and right-hand side of equation (6.28), respectively. What we want to prove is $[\![LHS]\!] = [\![RHS]\!]$. But as explained in the proof of Theorem 6.5.1, we need to work with the semi-classical semantics, and show that $\lfloor\!\lfloor LHS\rfloor\!\rfloor = \lfloor\!\lfloor RHS\rfloor\!\rfloor$. Assume that $\lfloor\!\lfloor S_i\rfloor\!\rfloor$ is the operator-valued function over Δ_i such that

$$\lfloor\!\lfloor S_i\rfloor\!\rfloor(\delta_i) = F_{i\delta_i}$$

for each $\delta_i \in \Delta_i$ $(1 \le i \le n)$. We write:

$$S \equiv \ \mathbf{qif}\ (\square i \cdot U_q^{\dagger}|i\rangle \to S_i)\ \mathbf{fiq}.$$

Then for any state $|\psi\rangle = \sum_{i=1}^{n}|i\rangle|\psi_i\rangle$, where $|\psi_i\rangle \in \mathcal{H}_V$ $(1 \le i \le n)$, and $V = \bigcup_{i=1}^{n} qvar(S_i)$, we have:

$$\begin{aligned}
\lfloor\!\lfloor S\rfloor\!\rfloor(\oplus_{i=1}^{n}\delta_i)|\psi\rangle &= \lfloor\!\lfloor S\rfloor\!\rfloor(\oplus_{i=1}^{n}\delta_i)\left[\sum_{i=1}^{n}\left(\sum_{j=1}^{n}U_{ij}(U_q^{\dagger}|j\rangle)\right)|\psi_i\rangle\right] \\
&= \lfloor\!\lfloor S\rfloor\!\rfloor(\oplus_{i=1}^{n}\delta_i)\left[\sum_{j=1}^{n}(U_q^{\dagger}|j\rangle)\left(\sum_{i=1}^{n}U_{ij}|\psi_i\rangle\right)\right] \\
&= \sum_{j=1}^{n}\left(\prod_{k\neq j}\lambda_{k\delta_k}\right)(U_q^{\dagger}|j\rangle)F_{j\delta_j}\left(\sum_{i=1}^{n}U_{ij}|\psi_i\rangle\right),
\end{aligned}$$

where $\lambda_{k\delta_k}$'s are defined by equation (6.15). Then it holds that

$$\begin{aligned}
[\![RHS]\!](\oplus_{i=1}^n \delta_i)|\psi\rangle &= U_{\overline{q}}([\![S]\!](\oplus_{i=1}^n \delta_i)|\psi\rangle) \\
&= \sum_{j=1}^n \left(\prod_{k\neq j} \lambda_{k\delta_k}\right) |j\rangle F_{j\delta_j} \left(\sum_{i=1}^n U_{ij}|\psi_i\rangle\right) \\
&= [\![S]\!](\oplus_{i=1}^n \delta_i)\left[\sum_{j=1}^n |j\rangle \left(\sum_{i=1}^n U_{ij}|\psi_i\rangle\right)\right] \\
&= [\![S]\!](\oplus_{i=1}^n \delta_i)\left[\sum_{i=1}^n \left(\sum_{j=1}^n U_{ij}|j\rangle\right)|\psi_i\rangle\right] \\
&= [\![S]\!](\oplus_{i=1}^n \delta_i)\left(\sum_{i=1}^n (U_{\overline{q}}|i\rangle)|\psi_i\rangle\right) \\
&= [\![LHS]\!](\oplus_{i=1}^n \delta_i)|\psi\rangle.
\end{aligned}$$

So, we complete the proof of equation (6.28).

Now we turn to prove the harder part, namely equation (6.29). The basic idea is to use equation (6.28) that we just proved to prove the more general equation (6.29). What we need to do is to turn the general "coin" program S in equation (6.29) into a special "coin" program, which is a unitary transformation. The technique that we used before to deal with quantum operations is always the Kraus operator-sum representation. Here, however, we have to employ the system-environment model of quantum operations (see Theorem 2.1.1). Since $[\![S]\!]$ is a quantum operation in $\mathcal{H}_{\overline{q}}$, there must be a family of quantum variables $\overline{r}$, a pure state $|\varphi_0\rangle \in \mathcal{H}_{\overline{r}}$, a unitary operator U in $\mathcal{H}_{\overline{q}} \otimes \mathcal{H}_{\overline{r}}$, and a projection operator K onto some closed subspace $\mathcal{K}$ of $\mathcal{H}_{\overline{r}}$ such that

$$[\![S]\!](\rho) = tr_{\mathcal{H}_{\overline{r}}}(KU(\rho \otimes |\varphi_0\rangle\langle\varphi_0|)U^\dagger K) \tag{6.43}$$

for all density operators ρ in $\mathcal{H}_{\overline{q}}$. We choose an orthonormal basis of $\mathcal{K}$ and then extend it to an orthonormal basis $\{|j\rangle\}$ of $\mathcal{H}_{\overline{r}}$. Define pure states $|\psi_{ij}\rangle = U^\dagger|ij\rangle$ for all i, j and programs

$$Q_{ij} \equiv \begin{cases} S_i & \text{if } |j\rangle \in \mathcal{K}, \\ \textbf{abort} & \text{if } |j\rangle \notin \mathcal{K}. \end{cases}$$

Then by a routine calculation we have:

$$[\![\textbf{qif}\ (\square i, j \cdot |ij\rangle \to Q_{ij})\ \textbf{fiq}]\!](\sigma) = [\![\textbf{qif}\ (\square i \cdot |i\rangle \to S_i)\ \textbf{fiq}]\!](K\sigma K) \tag{6.44}$$

for any $\sigma \in \mathcal{H}_{\overline{q}\cup\overline{r}\cup V}$, where

$$V = \bigcup_{i=1}^{n} qvar(S_i).$$

We now write *RHS* for the right-hand side of equation (6.29). Then we have:

$$\begin{aligned}
[\![RHS]\!](\rho) &= tr_{\mathcal{H}_{\overline{r}}}\left([\![\mathbf{qif}\ (\square i,j \cdot U^{\dagger}|ij\rangle \rightarrow Q_{ij})\ \mathbf{fiq};\ U[\overline{q},\overline{r}]]\!](\rho \otimes |\varphi_0\rangle\langle\varphi_0|\right) \\
&= tr_{\mathcal{H}_{\overline{r}}}\left(\left[\!\!\left[[U[\overline{q},\overline{r}]] \left(\bigoplus_{i,j} |ij\rangle \rightarrow Q_{ij}\right)\right]\!\!\right] (\rho \otimes |\varphi_0\rangle\langle\varphi_0|)\right) \\
&= tr_{\mathcal{H}_{\overline{r}}}\left([\![\mathbf{qif}\ (\square i,j \cdot |ij\rangle \rightarrow Q_{ij})\ \mathbf{fiq}]\!](U(\rho \otimes |\varphi_0\rangle\langle\varphi_0|)U^{\dagger})\right) \\
&= tr_{\mathcal{H}_{\overline{r}}}[\![\mathbf{qif}\ (\square i \cdot |i\rangle \rightarrow S_i)\ \mathbf{fiq}]\!](KU(\rho \otimes |\varphi_0\rangle\langle\varphi_0|)U^{\dagger}K) \\
&= [\![\mathbf{qif}\ (\square i \cdot |i\rangle \rightarrow S_i)\ \mathbf{fiq}]\!](tr_{\mathcal{H}_{\overline{r}}}(KU(\rho \otimes |\varphi_0\rangle\langle\varphi_0|)U^{\dagger}K)) \\
&= [\![\mathbf{qif}\ (\square i \cdot |i\rangle \rightarrow S_i)\ \mathbf{fiq}]\!]([\![S]\!](\rho)) \\
&= \left[\!\!\left[[S] \left(\bigoplus_{i} |i\rangle \rightarrow S_i\right)\right]\!\!\right] (\rho)
\end{aligned}$$

for all density operators ρ in $\mathcal{H}_{\overline{q}}$. Here, the second equality is obtained by using equation (6.28), the fourth equality comes from (6.44), the fifth equality holds because

$$\overline{r} \cap qvar(\mathbf{qif}\ (\square i \cdot |i\rangle \rightarrow S_i)\ \mathbf{fiq}) = \emptyset,$$

and the sixth equality follows from equation (6.43). Therefore, equation (6.29) is proved.

Proof of Theorem 6.6.1 and Theorem 6.6.3. The proof of Theorem 6.6.1 is similar to but simpler than the proof of Theorem 6.6.3. So, here we only prove Theorem 6.6.3.

(1) Clause (i) is immediate from Theorem 6.5.1.
(2) To prove clause (ii), we write:

$$\begin{aligned}
Q &\equiv \mathbf{qif}\ [\overline{q}](\square i \cdot |i\rangle \rightarrow S_i)\ \mathbf{fiq}, \\
R &\equiv \mathbf{qif}\ [\overline{q}](\square i \cdot |i\rangle \rightarrow S_{\tau(i)})\ \mathbf{fiq}.
\end{aligned}$$

By definition, we have $LHS = S; R$ and

$$RHS = S; U_{\tau}[\overline{q}]; Q; U_{\tau^{-1}}[\overline{q}].$$

So, it suffices to show that $R \equiv U_{\tau}[\overline{q}]; Q; U_{\tau^{-1}}[\overline{q}]$. Again, we first need to deal with the semi-classical semantics of the two sides of this equality. Assume that $\lfloor S_i \rfloor$ is the operator-valued function over Δ_i with

$$\lfloor S_i \rfloor(\delta_i) = E_{i\delta_i}$$

for each $\delta_i \in \Delta_i$ $(1 \le i \le n)$. For each state $|\Psi\rangle \in \mathcal{H}_{\overline{q}\cup\bigcup_{i=1}^{n} qvar(S_i)}$, we can write

$$|\Psi\rangle = \sum_{i=1}^{n} |i\rangle|\psi_i\rangle$$

for some $|\psi_i\rangle \in \mathcal{H}_{\bigcup_{i=1}^{n} qvar(S_i)}$ $(1 \le i \le n)$. Then for any $\delta_1 \in \Delta_{\tau(1)}, \dots, \delta_n \in \Delta_{\tau(n)}$, it holds that

$$\begin{aligned}|\Psi_{\delta_1\dots\delta_n}\rangle &\triangleq [\![R]\!](\oplus_{i=1}^{n}\delta_i)(|\Psi\rangle)\\ &= \sum_{i=1}^{n}\left(\prod_{k\neq i}\mu_{k\delta_k}\right)|i\rangle(E_{\tau(i)\delta_i}|\psi_i\rangle),\end{aligned}$$

where:

$$\mu_{k\delta_k} = \sqrt{\frac{trE^{\dagger}_{\tau(k)\delta_k}E_{\tau(k)\delta_k}}{\sum_{\theta_k\in\Sigma_{\tau(k)}} trE^{\dagger}_{\tau(k)\theta_k}E_{\tau(k)\theta_k}}} = \lambda_{\tau(k)\delta_k} \tag{6.45}$$

for every k and δ_k, and $\lambda_{i\sigma_i}$'s are defined by equation (6.15). On the other hand, we first observe:

$$|\Psi'\rangle \triangleq (U_\tau)_{\overline{q}}(|\Psi\rangle) = \sum_{i=1}^{n}|\tau(i)\rangle|\psi_i\rangle = \sum_{j=1}^{n}|j\rangle|\psi_{\tau^{-1}(j)}\rangle.$$

Then for any $\delta_1 \in \Delta_1, \dots, \delta_n \in \Delta_n$, it holds that

$$\begin{aligned}|\Psi''_{\delta_1\dots\delta_n}\rangle &\triangleq [\![Q]\!]\left(\bigoplus_{i=1}^{n}\delta_i\right)(|\Psi'\rangle)\\ &= \sum_{j=1}^{n}\left(\prod_{l\neq j}\lambda_{l\delta_{\tau^{-1}(l)}}\right)|j\rangle(E_{j\delta_{\tau^{-1}(j)}}|\psi_{\tau^{-1}(j)}\rangle)\\ &= \sum_{i=1}^{n}\left(\prod_{k\neq i}\lambda_{\tau(k)\delta_k}\right)|\tau(i)\rangle(E_{\tau(i)\delta_i}|\psi_i\rangle).\end{aligned}$$

Furthermore, we have:

$$(U_{\tau^{-1}})_{\overline{q}}(|\Psi''_{\delta_1\dots\delta_n}\rangle) = \sum_{i=1}^{n}\left(\prod_{k\neq i}\lambda_{\tau(k)\delta_k}\right)|i\rangle(E_{\tau(i)\delta_i}|\psi_i\rangle).$$

Consequently, we can compute the purely quantum semantics:

$$\begin{aligned}
[\![U_\tau[\overline{q}];Q;U_{\tau^{-1}}[\overline{q}]]\!](|\Psi\rangle\langle\Psi|) &= [\![Q;U_{\tau^{-1}}[\overline{q}](|\Psi'\rangle\langle\Psi'|)]\!]\\
&= (U_{\tau^{-1}})_{\overline{q}}\left(\sum_{\delta_1,\dots,\delta_n}|\Psi''_{\delta_1\dots\delta_n}\rangle\langle\Psi''_{\delta_1\dots\delta_n}|\right)(U_\tau)_{\overline{q}} \qquad (6.46)\\
&= \sum_{\delta_1,\dots,\delta_n}|\Psi_{\delta_1\dots\delta_n}\rangle\langle\Psi_{\delta_1\dots\delta_n}| = [\![R]\!](|\Psi\rangle\langle\Psi|).
\end{aligned}$$

Here, the third equality comes from equation (6.45) and the fact that τ is one-onto-one, and thus $\tau^{-1}(j)$ traverses over $1,\dots,n$ as j does. Therefore, it follows from equation (6.46) and spectral decomposition that

$$[\![R]\!](\rho) = [\![U_\tau[\overline{q}];Q;U_{\tau^{-1}}[\overline{q}]]\!](\rho)$$

for any density operator ρ in $\mathcal{H}_{\overline{q}\cup\bigcup_{i=1}^n qvar(S_i)}$, and we complete the proof of clause (ii).

(3) To prove clause (iii), we write:

$$\begin{aligned}
X_i &\equiv \textbf{qif}\ (\square j_i\cdot|j_i\rangle\rightarrow R_{ij_i})\ \textbf{fiq},\\
Y_i &\equiv [Q_i]\left(\bigoplus_{j_i=1}^{n_i}|j_i\rangle\rightarrow R_{ij_i}\right)
\end{aligned}$$

for every $1\le i\le m$, and we further put:

$$\begin{aligned}
X &\equiv \textbf{qif}\ (\square i\cdot|i\rangle\rightarrow Y_i)\ \textbf{fiq},\\
T &\equiv \textbf{qif}\ (\square i\cdot|i\rangle\rightarrow Q_i)\ \textbf{fiq},\\
Z &\equiv \textbf{qif}\ (\overline{\alpha})(\square i,j_i\in\Delta\cdot|i,j_i\rangle\rightarrow R_{ij_i})\ \textbf{fiq}.
\end{aligned}$$

Then by the definition of quantum choice we have $LHS = S;X$ and $RHS = S;T;Z$. So, it suffices to show that $X\equiv T;Z$. To do this, we consider the semi-classical semantics of the involved programs. For each $1\le i\le m$, and for each $1\le j_i\le n_i$, we assume:

- $\lfloor\!\lfloor Q_i\rfloor\!\rfloor$ is the operator-valued function over Δ_i such that

$$\lfloor\!\lfloor Q_i\rfloor\!\rfloor(\delta_i) = F_{i\delta_i}$$

  for every $\delta_i\in\Delta_i$; and
- $\lfloor\!\lfloor R_{ij_i}\rfloor\!\rfloor$ is the operator-valued function over Σ_{ij_i} such that

$$\lfloor\!\lfloor R_{ij_i}\rfloor\!\rfloor(\sigma_{ij_i}) = E_{(ij_i)\sigma_{ij_i}}$$

  for every $\sigma_{ij_i}\in\Sigma_{ij_i}$.

We also assume that state $|\Psi\rangle = \sum_{i=1}^{m} |i\rangle|\Psi_i\rangle$ where each $|\Psi_i\rangle$ is further decomposed into

$$|\Psi_i\rangle = \sum_{j_i=1}^{n_i} |j_i\rangle|\psi_{ij_i}\rangle$$

with $|\psi_{ij_i}\rangle \in \mathcal{H}_{\bigcup_{j_i=1}^{n_i} qvar(R_{ij_i})}$ for every $1 \leq i \leq m$ and $1 \leq j_i \leq n_i$. To simplify the presentation, we use the abbreviation $\overline{\sigma}_i = \oplus_{j_i=1}^{n_i}\sigma_{ij_i}$. Now we compute the semi-classical semantics of program Y_i:

$$\begin{aligned}
\llbracket Y_i \rrbracket(\delta_i\overline{\sigma}_i)|\Psi_i\rangle &= \llbracket X_i \rrbracket(\overline{\sigma}_i)(\llbracket Q_i \rrbracket(\delta_i)|\Psi_i\rangle) \\
&= \llbracket X_i \rrbracket(\overline{\sigma}_i)\left(\sum_{j_i=1}^{n_i}\left(F_{i\delta_i}|j_i\rangle\right)|\psi_{ij_i}\rangle\right) \\
&= \llbracket X_i \rrbracket(\overline{\sigma}_i)\left[\sum_{j_i=1}^{n_i}\left(\sum_{l_i=1}^{n_i}\langle l_i|F_{i\delta_i}|j_i\rangle|l_i\rangle\right)|\psi_{ij_i}\rangle\right] \\
&= \llbracket X_i \rrbracket(\overline{\sigma}_i)\left[\sum_{l_i=1}^{n_i}|l_i\rangle\left(\sum_{j_i=1}^{n_i}\langle l_i|F_{i\delta_i}|j_i\rangle|\psi_{ij_i}\rangle\right)\right] \\
&= \sum_{l_i=1}^{n_i}\left[\Lambda_{il_i}\cdot|l_i\rangle\left(\sum_{j_i=1}^{n_i}\langle l_i|F_{i\delta_i}|j_i\rangle E_{(il_i)\sigma_{il_i}}|\psi_{ij_i}\rangle\right)\right]
\end{aligned} \tag{6.47}$$

where the coefficients:

$$\Lambda_{il_i} = \prod_{l\neq l_i}\lambda_{(il)\sigma_{il}},$$

$$\lambda_{(il)\sigma_{il}} = \sqrt{\frac{trE^{\dagger}_{(il)\sigma_{il}}E_{(il)\sigma_{il}}}{\sum_{k=1}^{n_i} trE^{\dagger}_{(ik)\sigma_{ik}}E_{(ik)\sigma_{ik}}}}$$

for each $1 \leq l \leq n_i$. Then using equation (6.47), we can further compute the semi-classical semantics of program X:

$$\begin{aligned}
\llbracket X \rrbracket(\oplus_{i=1}^{m}(\delta_i\overline{\sigma}_i))|\Psi\rangle &= \sum_{i=1}^{m}\left(\Gamma_i\cdot|i\rangle\llbracket Y_i \rrbracket(\delta_i\overline{\sigma}_i)|\Psi_i\rangle\right) \\
&= \sum_{i=1}^{m}\sum_{l_i=1}^{n_i}\left[\Gamma_i\cdot\Lambda_{il_i}\cdot|il_i\rangle\left(\sum_{j_i=1}^{n_i}\langle l_i|F_{i\delta_i}|j_i\rangle E_{(il_i)\sigma_{il_i}}|\psi_{ij_i}\rangle\right)\right]
\end{aligned} \tag{6.48}$$

where:

$$\Gamma_i = \prod_{h\neq i} \gamma_{h\overline{\sigma}_h},$$

$$\gamma_{i\overline{\sigma}_i} = \sqrt{\frac{tr[\![Y_i]\!](\delta_i\overline{\sigma}_i)^\dagger[\![Y_i]\!](\delta_i\overline{\sigma}_i)}{\sum_{h=1}^m tr[\![Y_h]\!](\delta_h\overline{\sigma}_h)^\dagger[\![Y_h]\!](\delta_h\overline{\sigma}_h)}} \tag{6.49}$$

for every $1 \leq i \leq m$. On the other hand, we can compute the semi-classical semantics of program T:

$$\begin{aligned}
[\![T]\!](\oplus_{i=1}^m\delta_i)|\Psi\rangle &= [\![T]\!](\oplus_{i=1}^m\delta_i)\left(\sum_{i=1}^m |i\rangle|\Psi_i\rangle\right)\\
&= \sum_{i=1}^m \left(\Theta_i\cdot|i\rangle F_{i\delta_i}|\Psi_i\rangle\right)\\
&= \sum_{i=1}^m \left[\Theta_i\cdot|i\rangle\left(\sum_{j_i=1}^{n_i}(F_{i\delta_i}|j_i\rangle)|\psi_{ij_i}\rangle\right)\right]\\
&= \sum_{i=1}^m \left[\Theta_i\cdot|i\rangle\left(\sum_{j_i=1}^{n_i}\left(\sum_{l_i=1}^{n_i}\langle l_i|F_{i\delta_i}|j_i\rangle|l_i\rangle\right)|\psi_{ij_i}\rangle\right)\right]\\
&= \sum_{i=1}^m\sum_{l_i=1}^{n_i}\left[\Theta_i\cdot|il_i\rangle\left(\sum_{j_i=1}^{n_i}\langle l_i|F_{i\delta_i}|j_i\rangle|\psi_{ij_i}\rangle\right)\right]
\end{aligned}$$

where:

$$\Theta_i = \prod_{h\neq i}\theta_{h\delta_h},$$

$$\theta_{i\delta_i} = \sqrt{\frac{trF_{i\delta_i}^\dagger E_{i\delta_i}}{\sum_{h=1}^m trF_{h\delta_h}^\dagger F_{h\delta_h}}}$$

for every $1 \leq i \leq m$. Consequently, we obtain the semi-classical semantics of program $T;Z$:

$$\begin{aligned}
[\![T;Z]\!]\left(\left(\oplus_{i=1}^m\delta_i\right)\left(\oplus_{i=1}^m\overline{\sigma}_i\right)\right)|\Psi\rangle &= [\![Z]\!]\left(\oplus_{i=1}^m\overline{\sigma}_i\right)\left([\![T]\!]\left(\oplus_{i=1}^m\delta_i\right)|\Psi\rangle\right)\\
&= [\![Z]\!]\left(\oplus_{i=1}^m\oplus_{l_i=1}^{n_i}\sigma_{ij_i}\right)\left(\sum_{i=1}^m\sum_{l_i=1}^{n_i}\left[\Theta_i\cdot|il_i\rangle\left(\sum_{j_i=1}^{n_i}\langle l_i|F_{i\delta_i}|j_i\rangle|\psi_{ij_i}\rangle\right)\right]\right)\\
&= \sum_{i=1}^m\sum_{l_i=1}^{n_i}\left[\alpha^{il_i}_{\{\sigma_{jk_j}\}_{(j,k_j)\neq(i,l_i)}}\cdot\Theta_i\cdot|il_i\rangle\left(\sum_{j_i=1}^{n_i}\langle l_i|F_{i\delta_i}|j_i\rangle E_{(il_i)\sigma_{il_i}}|\psi_{ij_i}\rangle\right)\right].
\end{aligned} \tag{6.50}$$

By comparing equations (6.48) and (6.50), we see that it suffices to take

$$\alpha^{il_i}_{\{\sigma_{jk_j}\}_{(j,k_j)\neq(i,l_i)}} = \frac{\Gamma_i \cdot \Delta_{il_i}}{\Theta_i} \tag{6.51}$$

for all i, l_i and $\{\sigma_{jk_j}\}_{(j,k_j)\neq(i,l_i)}$. What remains to prove is the normalization condition:

$$\sum_{\{\sigma_{jk_j}\}_{(j,k_j)\neq(i,l_i)}} \left|\alpha^{il_i}_{\{\sigma_{jk_j}\}_{(j,k_j)\neq(i,l_i)}}\right|^2 = 1. \tag{6.52}$$

To do this, we first compute coefficients $\gamma_{i\overline{\sigma}_i}$. Let $\{|\varphi\rangle\}$ be an orthonormal basis of $\mathcal{H}_{\bigcup_{j_i=1}^{n_i} qvar(R_{ij_i})}$. Then we have:

$$G_{\varphi j_i} \overset{\triangle}{=} [\![Y_i]\!](\delta_i\overline{\sigma}_i)|\varphi\rangle|j_i\rangle = \sum_{l_i=1}^{n_i} \Lambda_{il_i} \cdot \langle l_i|F_{i\delta_i}|j_i\rangle E_{(il_i)\sigma_{il_i}}|\varphi\rangle|l_i\rangle.$$

It follows that

$$\begin{aligned} G^\dagger_{\varphi j_i} G_{\varphi j_i} &= \sum_{l_i,l_i'=1}^{n_i} \Lambda_{il_i} \cdot \Lambda_{il_i'} \langle j_i|F^\dagger_{i\delta_i}|l_i\rangle\langle l_i'|F_{i\delta_i}|j_i\rangle\langle\varphi|E^\dagger_{(il_i)\sigma_{il_i}} E_{(il_i')\sigma_{il_i'}}|\varphi\rangle\langle l_i|l_i'\rangle \\ &= \sum_{l_i=1}^{n_i} \Lambda^2_{il_i} \cdot \langle j_i|F^\dagger_{i\delta_i}|l_i\rangle\langle l_i|F_{i\delta_i}|j_i\rangle\langle\varphi|E^\dagger_{(il_i)\sigma_{il_i}} E_{(il_i)\sigma_{il_i}}|\varphi\rangle. \end{aligned}$$

Furthermore, we obtain:

$$\begin{aligned} tr[\![Y_i]\!](\delta_i\overline{\sigma}_i)^\dagger[\![Y_i]\!](\delta_i\overline{\sigma}_i) &= \sum_{\varphi,j_i} G^\dagger_{\varphi j_i} G_{\varphi j_i} \\ &= \sum_{l_i=1}^{n_i} \Lambda^2_{il_i} \cdot \left(\sum_{j_i} \langle j_i|F^\dagger_{i\delta_i}|l_i\rangle\langle l_i|F_{i\delta_i}|j_i\rangle\right)\left(\sum_{\varphi}\langle\varphi|E^\dagger_{(il_i)\sigma_{il_i}} E_{(il_i)\sigma_{il_i}}|\varphi\rangle\right) \\ &= \sum_{l_i=1}^{n_i} \Lambda^2_{il_i} \cdot tr(F^\dagger_{i\delta_i}|l_i\rangle\langle l_i|F_{i\delta_i}) tr(E^\dagger_{(il_i)\sigma_{il_i}} E_{(il_i)\sigma_{il_i}}). \end{aligned} \tag{6.53}$$

Now a routine but tedious calculation yields equation (6.52) through substituting equation (6.53) into (6.49) and then substituting equations (6.49) and (6.51) into (6.52).

(4) Finally, we prove clause (iv). To prove the first equality, we write:

$$\begin{aligned} X &\equiv \textbf{qif}\ (\square i \cdot |i\rangle \to S_i)\ \textbf{fiq}, \\ Y &\equiv \textbf{qif}\ (\overline{\alpha})(\square i \cdot |i\rangle \to (S_i; Q))\ \textbf{fiq}. \end{aligned}$$

Then by definition we have $LHS = S; X; Q$ and $RHS = S; Y$. So, it suffices to show that $X; Q =_{CF} Y$. Suppose that

$$\llbracket S_i \rrbracket(\sigma_i) = E_{i\sigma_i}$$

for every $\sigma_i \in \Delta(S_i)$ and $\llbracket Q \rrbracket(\delta) = F_\delta$ for every $\delta \in \Delta(Q)$, and suppose that

$$|\Psi\rangle = \sum_{i=1}^{n} |i\rangle|\psi_i\rangle$$

where $|\psi_i\rangle \in \mathcal{H}_{\bigcup_i qvar(S_i)}$ for all i. Then it holds that

$$\begin{aligned}\llbracket X; Q \rrbracket((\oplus_{i=1}^{n}\sigma_i)\delta)|\Psi\rangle &= \llbracket Q \rrbracket(\delta)(\llbracket X \rrbracket(\oplus_{i=1}^{n}\sigma_i)|\Psi\rangle) \\ &= F_\delta\left(\sum_{i=1}^{n}\Lambda_i|i\rangle(E_{i\sigma_i}|\psi_i\rangle)\right) \\ &= \sum_{i=1}^{n}\Lambda_i \cdot |i\rangle(F_\delta E_{i\sigma_i}|\psi_i\rangle)\end{aligned}$$

because $qvar(S) \cap qvar(Q) = \emptyset$, where:

$$\Lambda_i = \prod_{k \neq i}\lambda_{k\sigma_k},$$

$$\lambda_{i\sigma_i} = \sqrt{\frac{trE_{i\sigma_i}^{\dagger}E_{i\sigma_i}}{\sum_{k=1}^{n} trE_{k\sigma_k}^{\dagger}E_{k\sigma_k}}}. \tag{6.54}$$

Furthermore, we have:

$$\begin{aligned}&tr_{\mathcal{H}_{qvar(S)}}(\llbracket X; Q \rrbracket(|\Psi\rangle\langle\Psi|)) \\ &\quad = tr_{\mathcal{H}_{qvar(S)}}\left[\sum_{\{\sigma_i\},\delta}\sum_{i,j}\Lambda_i\Lambda_j \cdot |i\rangle\langle j|(F_\delta E_{i\sigma_i}|\psi_i\rangle\langle\psi_j|E_{j\sigma_j}^{\dagger}F_\delta^{\dagger})\right] \\ &\quad = \sum_{\{\sigma_i\},\delta}\sum_{i}\Lambda_i^2 \cdot F_\delta E_{i\sigma_i}|\psi_i\rangle\langle\psi_i|E_{i\sigma_i}^{\dagger}F_\delta^{\dagger}.\end{aligned} \tag{6.55}$$

On the other hand, we can compute the semi-classical semantics of Y:

$$\begin{aligned}\llbracket Y \rrbracket(\oplus_{i=1}^{n}\sigma_i\delta_i)|\Psi\rangle &= \sum_{i=1}^{n}\alpha_{\{\sigma_k,\delta_k\}_{k\neq i}}^{(i)} \cdot |i\rangle(\llbracket S_i; Q \rrbracket(\sigma_i\delta_i)|\psi_i\rangle) \\ &= \sum_{i=1}^{n}\alpha_{\{\sigma_k,\delta_k\}_{k\neq i}}^{(i)} \cdot |i\rangle(F_{\delta_i}E_{\sigma_i}|\psi_i\rangle).\end{aligned}$$

Furthermore, we obtain:

$$\begin{aligned}
&tr_{\mathcal{H}_{qvar(S)}}(\llbracket Y\rrbracket(|\Psi\rangle\langle\Psi|))\\
&= tr_{\mathcal{H}_{qvar(S)}}\left[\sum_{\{\sigma_i,\delta_i\}}\sum_{i,j}\alpha^{(i)}_{\{\sigma_k,\delta_k\}_{k\neq i}}(\alpha^{(j)}_{\{\sigma_l,\delta_l\}_{l\neq j}})^*\cdot|i\rangle\langle j|(F_{\delta_i}E_{i\sigma_i}|\psi_i\rangle\langle\psi_j|E^\dagger_{j\sigma_j}F^\dagger_{\delta_j})\right]\\
&= \sum_{\{\sigma_i\},\delta}\sum_i\left|\alpha^{(i)}_{\{\sigma_k,\delta_k\}_{k\neq i}}\right|^2\cdot F_{\delta_i}E_{i\sigma_i}|\psi_i\rangle\langle\psi_i|E^\dagger_{i\sigma_i}F^\dagger_{\delta_i}.
\end{aligned} \tag{6.56}$$

Comparing equations (6.55) and (6.56), we see that

$$tr_{\mathcal{H}_{qvar(S)}}(\llbracket X;Q\rrbracket(|\Psi\rangle\langle\Psi|)) = tr_{\mathcal{H}_{qvar(S)}}(\llbracket Y\rrbracket(|\Psi\rangle\langle\Psi|))$$

if we take

$$\alpha^{(i)}_{\{\sigma_k,\delta_k\}_{k\neq i}} = \frac{\Lambda_i}{\sqrt{|\Delta(Q)|}}$$

for all i, $\{\sigma_k\}$ and $\{\delta_k\}$. Since

$$qvar(S)\subseteq cvar(X;Q)\cup cvar(Y),$$

it follows that

$$tr_{\mathcal{H}_{cvar(X;Q)\cup cvar(Y)}}(\llbracket X;Q\rrbracket(|\Psi\rangle\langle\Psi|)) = tr_{\mathcal{H}_{cvar(X;Q)\cup cvar(Y)}}(\llbracket Y\rrbracket(|\Psi\rangle\langle\Psi|)).$$

Therefore, we can assert that

$$tr_{\mathcal{H}_{cvar(X;Q)\cup cvar(Y)}}(\llbracket X;Q\rrbracket(\rho)) = tr_{\mathcal{H}_{cvar(X;Q)\cup cvar(Y)}}(\llbracket Y\rrbracket(\rho))$$

for all density operators ρ by spectral decomposition. Thus, we have $X;Q\equiv_{CF} Y$, and the proof of the first equality of clause (iv) is completed.

For the special case where Q contains no measurements, $\Delta(Q)$ is a singleton, say $\{\delta\}$. We write:

$$Z\equiv\ \textbf{qif}\ (\square i\cdot|i\rangle\rightarrow(P_i;Q))\ \textbf{fiq}.$$

Then

$$\llbracket Z\rrbracket(\oplus_{i=1}^n\sigma_i\delta)|\Psi\rangle = \sum_{i=1}^n\left(\prod_{k\neq i}\theta_{k\sigma_k}\right)\cdot|i\rangle(F_\delta E_{\sigma_i}|\psi_i\rangle),$$

where:

$$\theta_{i\sigma_i} = \sqrt{\frac{trE^\dagger_{i\sigma_i}F^\dagger_\delta F_\delta E_{i\sigma_i}}{\sum_{k=1}^n trE^\dagger_{k\sigma_k}F^\dagger_\delta F_\delta E_{k\sigma_k}}} = \lambda_{i\sigma_i},$$

and $\lambda_{i\sigma_i}$ is given by equation (6.54), because $F_\delta^\dagger F_\delta$ is the identity operator. Consequently, $[\![X;Q]\!] = [\![Z]\!]$, and we complete the proof of the second equality of clause (iv).

6.10 BIBLIOGRAPHIC REMARKS

This chapter is mainly based on the draft paper [233]; an earlier version of [233] appeared as [232]. The examples presented in Subsection 6.7.1 were taken from recent physics literature: the unidirectional quantum walk was examined in [171]; the quantum walk with "coin tossing operator" depending on time and position was employed in [145] to implement quantum measurement; the quantum walk with three coin states was considered in [122]; the one-dimensional quantum walk driven by multiple coins was defined in [49]; and the quantum walk consisting of two walkers on a line sharing coins was introduced in [217].

- *GCL and its extensions*: The programming language QuGCL studied in this chapter is a quantum counterpart of Dijkstra's GCL (Guarded Command Language). The language GCL was originally defined in [74], but one can find an elegant and systematic presentation of GCL in [172]. The language pGCL for probabilistic programming was defined by introducing probabilistic choice into GCL; for a systematic exposition of probabilistic programming with probabilistic choice, we refer to [166]. A comparison between quantum choice and probabilistic choice was given in Section 6.5.

  Another quantum extension qGCL of GCL was defined in Sanders and Zuliani's pioneering paper [191]; see also [241]. qGCL was obtained by adding three primitives for quantum computation – initialization, unitary transformation, quantum measurement – into the probabilistic language pGCL. Note that the control flows of qGCL programs are always classical. QuGCL can also be seen as an extension of qGCL obtained by adding a quantum case statement (with quantum control flow).
- *Quantum control flow*: Quantum programs with quantum control flow were first considered by Altenkirch and Grattage [14], but the way of defining quantum control flow in this chapter is very different from that used in [14]. A careful discussion about the difference between the approach in [14] and ours can be found in [232,233]. Our approach was mainly inspired by the following line of research: a superposition of evolutions (rather than that of states) of a quantum system was considered by physicists Aharonov et al. [11] as early as 1990, and they proposed to introduce an external system in order to implement the superposition. The idea of using such an external "coin" system was rediscovered by Aharonov et al. and Ambainis et al. in defining quantum walks [9,19]. The fact that the shift operator S of a quantum walk can be seen as a quantum case statement and the single-step operator W as a quantum choice was noticed in [232,233] by introducing the single-direction shift operators S_i's.

It then motivated the design decision of quantum case statement and quantum choice as well as the quantum programming paradigm of superposition-of-programs.

Whenever no measurements are involved, then the semantics of a quantum case statement can be defined in terms of the guarded composition of unitary operators. However, defining the semantics of a quantum case statement in its full generality requires the notion of guarded composition of quantum operations, introduced in [232,233].

- *More related literature*: As pointed out in Subsection 6.3.1, guarded composition of unitary operators is essentially a quantum multiplexor introduced by Shende et al. [201] and discussed in Subsection 2.2.4. It was called a measuring operator by Kitaev et al. in [135].

  The quantum programming language Scaffold [3] supports quantum control primitives with the restriction that the code comprising the body of each module must be purely quantum (and unitary). Hence, its semantics only requires guarded composition of unitary operators, defined in Subsection 6.3.1. Recently, some very interesting discussions about quantum case statements were given by Bădescu and Panangaden [28]; in particular, they observed that quantum case statements are not monotone with respect to the Löwner order and thus not compatible with the semantics of recursion defined in [194].

  In recent years, there have been quite a few papers concerning superposition of quantum gates or more general quantum operations in the physics literature. Zhou et al. [240] proposed an architecture-independent technique for adding control to arbitrary unknown quantum operations and demonstrated this in a photonic system. This problem was further considered by Araújo et al. [22] and Friis et al. [91].

  The interesting idea of superposition of causal structures in quantum computation was first introduced by Chiribella et al. [55]. It was generalized by Araújo et al. [23] and implemented by Procopio et al. [181].