Quantum Control and General Recursion beyond the Unitary Case

Kathleen Barsse, Romain Péchoux and Simon Perdrix
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

The full paper is available at `https://arxiv.org/abs/2507.10466`

1 Context and Motivations

In quantum computing, the execution flow of operations acting on quantum states is generally carried out in a fixed or classically controlled manner, based on information such as the classical outcomes of quantum measurements. This is the "quantum data, classical control" paradigm, which is known to enable a speedup in comparison to classical algorithms. One step further, one can also consider the superposition of quantum processes themselves, depending on a quantum state. This notion is referred to as *coherent control* or *quantum control*. A fundamental example of coherent control is the *quantum case*. This process can be thought of as a quantum version of the classical "`if`" statement where the two branches can run in a superposition. Namely, in the case of unitary evolutions, the quantum case acts on a pair of control and target qubits as follows: if the control qubit is in state $|0\rangle$, a unitary U is applied to the target qubit, if the control qubit is in state $|1\rangle$, a unitary V is applied to the target qubit, otherwise U and V are applied in a coherent superposition depending on the state of the control qubit.

In comparison to quantum processes whose execution flow is fixed or classically controlled, models with coherent control are known to provide some advantages in computational complexity [6, 8, 2, 9] and communication tasks [7, 1, 10]. For this reason, the possibility of including coherent control in quantum programming languages is attracting a great deal of interest (see [12] for an overview).

The main challenge is to combine *quantum control* with the key features of a quantum programming language: *general recursion*, and the capability to handle not only unitary evolutions but also *measurements*. Notice that there are already solid contributions that combine two of these three features, that we briefly review below.

Without quantum control, it is well-known that completely positive maps, or quantum channels, can be used to represent both unitary evolutions and measurements. Moreover, Selinger [11] showed that, when equipped with the Löwner partial ordering, completely positive maps form a DCPO, thereby enabling the use of domain-theoretic fixed-point techniques for general recursion. Nevertheless, incorporating quantum control into this framework is far from straightforward, as the quantum case operation is incompatible with the Löwner ordering [4].

To circumvent these no-go results, a compelling line of research considers pure evolutions only, which entails dropping measurement capabilities and adopting quantum-controlled recursion. Indeed, whereas recursion usually relies on measurements to determine whether recursive calls should be made, one can consider quantum controlled recursion in which recursive calls are performed in superposition. However, suitable definitions of quantum recursion, such as a quantum while loop, involves not only to consider systems of infinite dimension, but also to define the outcomes of programs as limits of infinite computations [15, 14, 3]. This makes its practical significance questionable, as for instance the notion of termination is unclear in this context.

Finally, in the absence of general recursion, combining quantum control with measurement raises the issue of controlling general quantum operations (beyond the unitary case), which is non trivial. Indeed, although the quantum case is well defined on unitary maps, namely as $(U, V) \mapsto |0\rangle\langle 0| \otimes U + |1\rangle\langle 1| \otimes V$, its extension to completely positive maps is ill-defined [4]. In order to support control of non unitary operations, it is necessary to choose a different semantic domain. This is achieved in [17, 16, 4], with semantics relying on Kraus decompositions. Notice that in recent years, the question of coherently controlling arbitrary quantum operations (in the absence of general recursion) has been intensively studied in foundations of quantum mechanics, where it is pointed out that coherent control of quantum operations requires to specify additional information on the implementation of each controlled channel [10, 1]. Various frameworks have been introduced to this end, including *vacuum-extended operations* [10, 5], *routed channels* [13] and Kraus decompositions with input environment states [1].

In summary, the development of a full-fledged programming language that combines quantum control with general recursion and measurement has remained an open problem for over a decade, as existing frameworks supporting general recursion either lack quantum control or quantum measurement. Solving it is nevertheless essential to the development of high-level quantum programming languages in their full generality.

2 Contributions

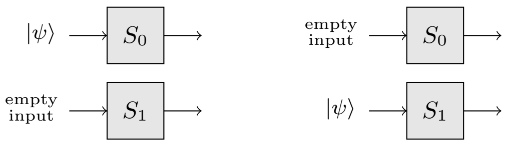
We solve the above longstanding problem by providing the first semantically well-behaved programming language allowing for measurements, recursion and quantum control beyond the unitary case. Quantum control is formulated as a quantum case primitive called "`qcase`", which we present as a quantum version of the classical conditional statement. The novelty of our programming language is the possibility for both coherent control of arbitrary quantum processes (i.e., not restricted to unitary maps) and recursion via a classically controlled while loop. The syntax is defined by the following grammar:

$$\begin{aligned} S, S_0, S_1 ::= \ & \texttt{skip} \mid \texttt{new qbit q} \mid \texttt{discard q} \mid \texttt{q}{*}{=}U \mid S_0; S_1 \\ & \mid \ \texttt{meas q}\ (0 \to S_0, 1 \to S_1) \mid \texttt{while q do}\ S \mid \ \texttt{qcase q}\ (0 \to S_0, 1 \to S_1) \end{aligned}$$

The syntax can be viewed as that of QPL [11] extended by a quantum case. `skip` is a no-op statement. `new qbit q` initializes the qubit variable `q` in the state $|0\rangle$, and `discard` traces out its argument. The statement $\texttt{q}{*}{=}U$ applies the single-qubit unitary gate U to `q`. $S_0; S_1$ applies S_0 then S_1 in a sequence. The statement $\texttt{meas q}\ (0 \to S_0, 1 \to S_1)$ performs a standard computational basis measurement on `q` and executes S_0 or S_1 depending on the outcome. The statement $\texttt{while q do}\ S$ is a classically-controlled loop: with each iteration, the control qubit `q` is measured in the computational basis, and the loop is exited once outcome 0 is obtained. The statement $\texttt{qcase q}\ (0 \to S_0, 1 \to S_1)$ executes S_0 and S_1 in a coherent superposition, depending on the state of the control qubit `q`. Importantly, qubit `q` is not measured during this process. This ensures that the resulting process is a superposition of the two possible statements rather than a probabilistic mixture.

In order to assign a formal meaning to programs, we construct an operational semantics, consisting of a probabilistic big-step transition system, and a denotational semantics. Despite the apparent issues surrounding the definition of the quantum case [4], this operation is physically meaningful. Intuitively, it requires some additional information that describes the behavior of the program in the absence of inputs. In the operational semantics, this additional information takes the form of a default transition for each probabilistic statement. In the denotational semantics, the additional information is expressed using the framework of *coherent quantum operations*, which take inspiration from vacuum-extended operations [10, 5]. Coherent quantum operations are shown to form a pointed DCPO allowing for domain-theoretic fixed-point techniques for general recursion in the presence of quantum control and arbitrary quantum operations. Hence, this enables us to define the semantics of the quantum case in a physically meaningful way, without limiting which operations are considered controllable, while ensuring the usual expected properties such as compositionality.

The operational semantics defines the evolution of an input state $|\psi\rangle$ induced by a program. To explain the semantics of `qcase`, consider the statement $\texttt{qcase q}\ (0 \to S_0, 1 \to S_1)$. We begin by examining the base cases. If the input has the form $|0\rangle_{\texttt{q}} \otimes |\psi\rangle$, since the control qubit is in state $|0\rangle$, the target input $|\psi\rangle$ is passed to S_0, and S_1 does not receive any input. And conversely, if the input has the form $|1\rangle_{\mathtt{q}} \otimes |\psi\rangle$, the target input $|\psi\rangle$ is passed to S_1, and S_0 does not receive any input. This is illustrated as follows:



No input on S_1 No input on S_0

The action on superpositions is obtained by linearity. Therefore, it is necessary to define the action of programs both on regular inputs and on the empty input. Given that the transition system is probabilistic, it suffices to assign to each program a default transition: when the input is empty, this is the transition that occurs. Accordingly, transitions will have the form $[S, |\psi\rangle]_\Gamma \xrightarrow{\nu} |\psi'\rangle$, for some program statement S, input state $|\psi\rangle$, output state $|\psi'\rangle$ and $\nu \in \{0,1\}$. The bit $\nu \in \{0,1\}$ indicates whether this transition is the default one, and the transition occurs with probability $\frac{\||\psi'\rangle\|^2}{\||\psi\rangle\|^2}$, if $|\psi\rangle \neq 0$ (probability 0 otherwise).

The denotational semantics is described using coherent quantum operations. A coherent quantum operations is a pair $(\mathcal{C}, F)$, where $\mathcal{C}$ is a quantum operation (i.e., a completely positive trace non-increasing linear map), and F is a linear map called a *transformation matrix*. Although it is not possible to define the quantum case operation on quantum operations only, the adjunction of a transformation matrix (interpreted as necessary information about the implementation of the quantum channel in [1, 10]) carries sufficient additional information for well-defined quantum control. Therefore, the denotational semantics assigns to each program a pair $(\mathcal{C}, F)$.

We prove the following main properties of the language:

- **Universality** (Theorem 5.1): every quantum operation can be implemented by a program in the language. Precisely, every quantum operation can be implemented by a coherent quantum operation, and every coherent quantum operations can be realized as the interpretation of some program. Moreover, approximate universality is shown when the set of built-in unitary gates is restricted to the Hadamard gate H and the T-gate.

- **Adequacy** (Theorem 6.1): the independently-defined operational and denotational semantics are adequate, meaning that they describe the same behavior of programs. Consequently, this links the default transition approach of the operational semantics to the coherent quantum operation approach of the denotational semantics.

- **Full abstraction** (Theorem 7.4): we introduce a notion of observational equivalence, which states that two programs are observationally equivalent if their probability of termination is the same whatever the context. This formalizes the idea that the observable behavior of two programs is indistinguishable. We show that the denotational semantics is fully abstract for observational equivalence. Hence, the denotational semantics exactly captures the observable behavior of programs.

References

[1] Alastair A. Abbott et al. “Communication through coherent control of quantum channels”. In: *Quantum* 4 (2020), p. 333.

[2] Mateus Araújo, Fabio Costa, and Časlav Brukner. “Computational advantage from quantum-controlled ordering of gates”. In: *Physical review letters* 113.25 (2014), p. 250402.

[3] Nicola Assolini and Alessandra Di Pierro. “A Denotational Semantics for Quantum Loops”. In: *arXiv preprint arXiv:2506.23320* (2025).

[4] Costin Bădescu and Prakash Panangaden. “Quantum Alternation: Prospects and Problems”. In: *Proceedings 12th International Workshop on Quantum Physics and Logic, QPL 2015, Oxford, UK, July 15-17, 2015.* 2015, pp. 33–42.

[5] Giulio Chiribella and Hlér Kristjánsson. “Quantum Shannon theory with superpositions of trajectories”. In: *Proceedings of the Royal Society A* 475.2225 (2019), p. 20180903.

[6] Timoteo Colnaghi et al. In: *Physics Letters A* 376.45 (2012), pp. 2940–2943.

[7] Daniel Ebler, Sina Salek, and Giulio Chiribella. “Enhanced communication with the assistance of indefinite causal order”. In: *Physical review letters* 120.12 (2018), p. 120502.

[8] Stefano Facchini and Simon Perdrix. “Quantum circuits for the unitary permutation problem”. In: *International Conference on Theory and Applications of Models of Computation.* Springer. 2015, pp. 324–331.

[9] Hlér Kristjánsson et al. “Exponential separation in quantum query complexity of the quantum switch with respect to simulations with standard quantum circuits”. In: *arXiv preprint arXiv:2409.18420* (2024).

[10] Hlér Kristjánsson et al. “Resource theories of communication”. In: *New Journal of Physics* 22.7 (2020), p. 073014.

[11] Peter Selinger. “Towards a quantum programming language”. In: *Mathematical Structures in Computer Science* 14.4 (2004), pp. 527–586.

[12] Benoît Valiron. “Semantics of quantum programming languages: Classical control, quantum control”. In: *Journal of Logical and Algebraic Methods in Programming* 128 (2022), p. 100790.

[13] Augustin Vanrietvelde, Hlér Kristjánsson, and Jonathan Barrett. “Routed quantum circuits”. In: *Quantum* 5 (2021), p. 503.

[14] Mingsheng Ying. *Foundations of quantum programming.* Elsevier, 2024.

[15] Mingsheng Ying. “Quantum recursion and second quantisation”. In: *arXiv preprint arXiv:1405.4443* (2014).

[16] Mingsheng Ying, Nengkun Yu, and Yuan Feng. “Alternation in quantum programming: from superposition of data to superposition of programs”. In: *arXiv preprint arXiv:1402.5172* (2014).

[17] Mingsheng Ying, Nengkun Yu, and Yuan Feng. “Defining quantum control flow”. In: *arXiv preprint arXiv:1209.4379* (2012).