

An Introduction to Measurement-Based Quantum Computation

Liam Miller

*Institute for Quantum Science and Technology and Department of Physics and Astronomy,
University of Calgary, Calgary, Alberta T2N 1N4, Canada*

Computation is often introduced through the lens of machines: logic gates, electronic circuits, and physical hardware. While these descriptions are essential for implementation, it can miss some of the underlying mathematical structure shared by all models of computation. At its core, computation is not about any particular device, but about how information is represented and transformed.

In this text, we take a structural approach and view computation as the composition of maps or functions acting on well-defined information spaces. This viewpoint allows classical and quantum computation to be discussed within a common mathematical language, with the main differences arising from the types of spaces involved and the transformations allowed on them.

The goal of this text is to provide a clear and mathematically grounded introduction to measurement-based quantum computation (MBQC). The emphasis is on understanding structure rather than hardware, and on using precise definitions without unnecessary technical overhead. Along the way, we will connect MBQC to ideas from tensor networks and symmetry-based descriptions of quantum states, highlighting how familiar tools from linear algebra naturally lead to this model of computation.

By approaching MBQC from this angle, we hope to make it feel less mysterious and more like a natural extension of ideas already present in classical and circuit-based quantum computation.

I. Classical Computation: A Bird's Eye View

Classical computation provides a natural and accessible starting point for exploring the mathematical framework that underlies computation. By examining classical computation in abstract terms, we can isolate the essential ingredients common to all computational models: the representation of information, the transformations acting on that information, and the way these transformations are combined to form algorithms.

In this section, we focus on these ingredients while deliberately setting aside implementation details. This allows us to describe classical computation in a way that will generalize naturally to quantum computation, where the same structural ideas reappear in a different mathematical setting.

A. Classical Information

In classical computation, information is *discrete* and can be modeled using *sets*. A single classical datum corresponds to an element of a set, with each element representing a distinct, mutually exclusive configuration of information (called a bit). The set as a whole provides a complete description of the informational landscape relevant to a given computational task, specifying *every-*

thing that could possibly be known about the data being processed.

B. Computations as Maps

Given such a set that provides a complete description of the informational possibilities relevant to a computational task, it is natural to describe classical computations as *maps between sets*. A deterministic classical computation acts by taking an input configuration from a specified set and transforming it into an output configuration according to a fixed rule. Mathematically, such a transformation is represented by a function

$$f : X \rightarrow Y,$$

where X denotes the set of possible input states and Y denotes the set of possible outputs.

To capture the full expressive power of classical computation, the input set X must in general be infinite. This reflects the fact that universal models of computation, such as the Turing machine, operate on unbounded inputs and may access arbitrarily large intermediate configurations. Finite input sets give rise only to limited computational models, whereas allowing X to be infinite permits the representation of all computable functions.

From this perspective, a computational task is fully characterized by the choice of input and output sets together with a function relating them. The function specifies *what* computation is being performed, while the sets determine the domain over which it is defined. An *algorithm* then provides a concrete realization of this function by decomposing it into a sequence of simpler, elementary transformations, together with rules for their composition and iteration.

II. Quantum Computing

The structural description developed above provides a natural starting point for quantum computation. The transition from classical to quantum computation does not alter the underlying notion of computation as a transformation of information, but rather changes the mathematical objects used to represent information and the class of transformations allowed.

In quantum computation, information is represented by vectors in complex Hilbert spaces. A quantum system is associated with a Hilbert space \mathcal{H} , and its state is described by a unit vector $|\psi\rangle \in \mathcal{H}$. For example, a single qubit is represented by the two-dimensional Hilbert space $\mathcal{H} \cong \mathbb{C}^2$, with a distinguished computational basis $\{|0\rangle, |1\rangle\}$.

Transformations of quantum information are described by linear maps acting on \mathcal{H} . In the ab-

sence of measurement, the evolution of a closed quantum system is given by unitary operators

$$U : \mathcal{H} \rightarrow \mathcal{H},$$

which preserve inner products and norm. More generally, quantum operations may include measurements, which are described by collections of linear operators satisfying appropriate completeness relations and give rise to probabilistic outcomes.

Despite these differences, the same structural roles identified in the classical setting remain present. A quantum computation is defined by a choice of information space (a Hilbert space), a collection of admissible transformations acting on that space, and rules for composing these transformations into algorithms. The essential distinction from classical computation lies not in the overall structure, but in the linear and probabilistic nature of the transformations involved.

From this perspective, quantum computation may be understood as a generalization of the classical framework in which sets are replaced by Hilbert spaces and functions by linear maps. This unified mathematical language will be used throughout the remainder of this text, providing a common foundation for discussing circuit-based quantum computation and measurement-based quantum computation within the same structural framework.