On Circuit Description Languages, Indexed Monads, and Resource Analysis

This paper presents a new denotational semantics for circuit description languages (CDLs) inspired by Quipper and its formal core calculi in the Proto-Quipper family. The proposed approach is based on indexed monads and is designed to clearly separate the value produced by a program from the quantum circuit generated as a side effect during its execution.

Traditional semantic models for Proto-Quipper rely heavily on presheaf constructions, where program values and circuits are intertwined into a single mathematical object. While expressive, this coupling makes it difficult to reason modularly about circuit properties such as size, width, or resource usage. In particular, it limits the ability to justify advanced type systems that aim to control or bound the resources required by generated circuits.

To address this issue, the paper introduces a monad-based framework in which circuit construction is modeled as an indexed monad. Each program is interpreted as producing both a value and a circuit, but these two components are kept explicitly separate. This separation enables clearer semantic reasoning and supports richer forms of effect typing that capture quantitative properties of circuits.

A revised calculus, called Proto-Quipper-C, is presented. It extends Proto-Quipper-M by annotating function types with explicit information about the circuit resources captured by closures. This modification is essential for determining the interface of circuits produced by higher-order functions and enables a sound monadic interpretation.

The denotational semantics of Proto-Quipper-C is defined using a parameterized Freyd category induced by the circuit indexed monad. The model is shown to be sound and computationally adequate with respect to the operational semantics. Moreover, the framework naturally generalizes to effect-typed systems, allowing circuit metrics such as size and width to be tracked compositionally, even in the presence of simple circuit optimizations.

Finally, the paper discusses possible extensions of the framework to dependent types, showing how families constructions can be used to enrich the semantics further. Overall, the work provides a flexible and conceptually clean foundation for reasoning about resource usage in quantum circuit description languages.