Meet Silq – The New High-level Programming Language For Quantum Computers

22/06/2020

AMBIKA CHOUDHURY

A Technical Journalist who loves writing about Machine Learning and…

Intro

- Recently, developers from ETH Zurich, Switzerland introduced the first-ever high-level programming language for quantum computers, known as Silq. The language addresses various challenges of quantum languages, such as unintuitive and cluttered code by supporting safe and automatic uncomputation.
- In order to gain an advantage over traditional algorithms, researchers have been working on quantum computers and algorithms for a few decades now.

Behind the Language

- Silq is a high-level programming language for quantum computing that is designed to extract from low-level implementation details of quantum algorithms.
- The technical novelty of this language is a quantum type system that captures important aspects of quantum computations and enables safe and automatic computation, which is, a fundamental challenge in existing quantum languages.
- According to the developers, Silq is the first quantum language with a strong static type system to provide intuitive semantics, i.e. if a program type-checks, its semantics follows an intuitive recipe that simply drops temporary values. This language allows expressing quantum algorithms more safely and concisely than existing quantum programming languages.

Why Use Silq

- Quantum computations often produce temporary values, and removing such values from consideration induces an implicit measurement collapsing the state of quantum computing. To remove the temporary values from consideration without inducing an implicit measurement, algorithms in existing languages must explicitly uncompute all the temporary values.
- This results in a significant gap from quantum to classical languages and is a major roadblock preventing the adoption of quantum languages, as the implicit side-effects resulting from uncomputation mistakes such as silently dropping temporary values are highly unintuitive.
- This is where Silq language comes into play. This language bridges this gap by automatically uncomputing temporary values. The type system of Silq exploits a fundamental pattern in quantum algorithms, stating that uncomputation can be done safely if: –
- The original evaluation of the uncomputed value can be described classically and
- The variables used to evaluate it are preserved and can thus be leveraged for uncomputation.

Benefits of Silq

- Silq's main advantage over existing quantum languages is its safe and automatic uncomputation, enabled by its novel annotations const and qfree.
- Silq algorithms are shorter and simpler in nature.
- The language enables intuitive yet physical semantics and statically prevents errors that are not detected in existing quantum languages.
- It modifies the program's quantum state according to an intuitive semantics that follows the laws of quantum physics.
- Silq avoids the notational overhead associated with languages that achieve lesser static safety in programs.

Installation

- The recommended way to install Silq is by using its Visual Studio Code Plugin (vscode plugin). This approach works for Linux, Mac, and Windows.
- To install Silq's vscode plugin
  - Open *vscode*
  - Open tab extensions *(Ctrl+Shift+X)*
  - Install *vscode-silq*
- Click here to know.

- Documentation and comparison to Q# which makes less code than Q#
- https://silq.ethz.ch/comparison

Wrapping Up

- According to the developers, in contrast to the existing quantum languages such as Q#, Quipper, ProjectQ, QWire, among others, this language supports a descriptive view of the quantum algorithms that expresses the high-level intent of the programmer. The key language features of this language can also be incorporated into existing languages such as QWire or Q#.
- The experimental evaluation of Silq demonstrated that the programs of this language are not only easier to read and write but also significantly shorter than equivalent programs in other quantum languages such as Q#, Quipper, etc. while using only half the number of quantum primitives. On average, Silq programs require 46% less code than Q#. In future work, the developers expect Silq to advance various tools central to programming quantum computers.
- Read the paper here. https://doi.org/10.1145/3385412.3386007
- Watch the video below-

END