Quantum Memory Management

Daily supervisor: Samuel Oslovich (s.o.oslovich@tudelft.nl)
Project supervisor: Stephanie Wehner

The quantum internet allows for exciting new applications that are unachievable using classical computers and the classical internet. When these applications are executed on quantum computers, they can be broken down into individual tasks consisting of local quantum gates, generating entanglement, and classical operations. This separation of tasks allows for multiple programs to be executed simultaneously, improving the program execution time. A key challenge in realizing this in near term quantum devices is how to manage the limited amount of quantum memory. Improper memory management can lead to decreased program performance or even deadlocks that prevent the program from executing entirely.

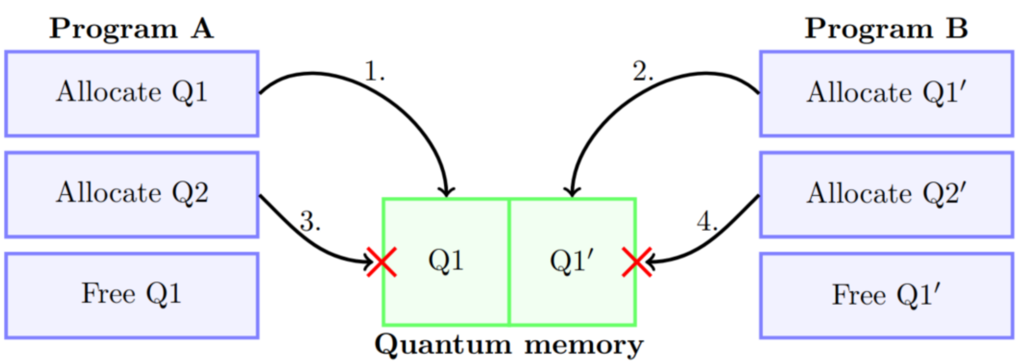

Figure 1: Tasks making up a part of programs A and B, are shown running on a quantum computer with two quantum memories. **1.** Program A allocates a single quantum memory (e.g., entanglement generation). **2.** Program B does the same. **3.** Program A attempts to allocate a second quantum memory (e.g., preparing a state to teleport). This fails since no memory is available. **4.** Program B also fails to allocate memory. Since both programs need to allocate more memory before freeing memory we encounter a deadlock, and neither program finishes executing.

A common solution in classical computing when programs run out of memory is to terminate them. Simply applying this strategy to quantum computers could have detrimental effects on performance. For example, establishing entanglement in near-term quantum computers is costly and prematurely terminating a program can cause the entanglement generation process to be repeated unnecessarily. This leads to interesting questions for memory management such as "which quantum program should we terminate?" and "how can we avoid terminating programs?"

The goal of this project is to research, implement, and evaluate different memory management strategies to understand their effects on program performance. This project also contributes to QIA, a collaboration with funding from the European Union with the mission to build a large-scale quantum network by 2029.

Project Goals:

- Investigate memory allocation strategies
- Implement (one or more) of these strategies in a simulated quantum operating system
- Evaluate the performance of memory allocation strategies

Valued Background:

The student should have a background in computer science and be familiar with programming in Python and using git for version control. Basic knowledge of operating systems is preferred but not required. Basic knowledge of quantum computing is appreciated but not required.