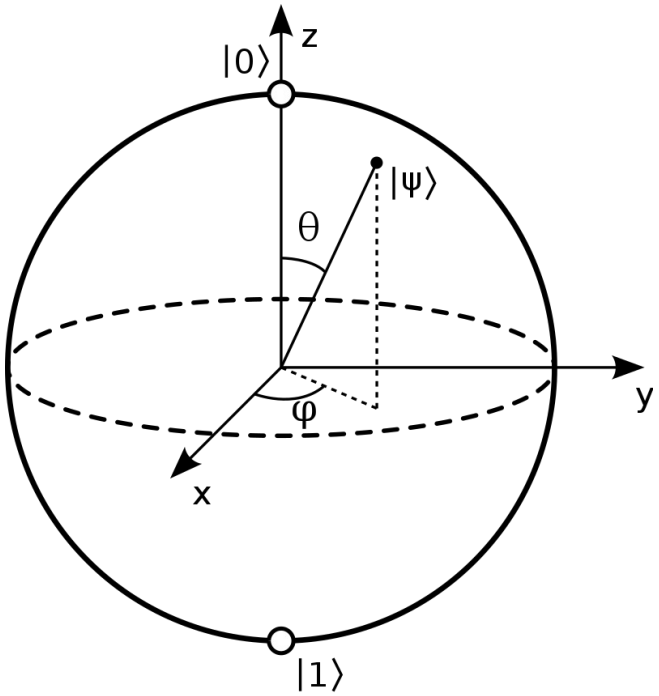
Bounded Verification of Quantum Programs



Quantum computation is one of “the” emerging fields of the last decades[2]. Promising super-polynomial speedup for certain problems this field of research has attracted huge attention. However writing quantum programs is way harder than traditional programs and thus more error prone. In this project the goal is to develop a methodology to verify bounded versions of quantum algorithms like the Shor-Algorithm (factoring of integers) using fully automatic bounded model checkers. A challenge will be to combine classical parts and quantum parts of the algorithms into one verification task in contrast to only verifying the quantum circuits. The underlying idea is to use a translation of the given algorithm into an existing programming language for which bounded model checkers already exist[2] using them to discharge the verification conditions. Tasks for this project will include:

1. Developing a translation between quantum circuits and a classical programming language
2. Developing a way to specify quantum algorithms
3. Implementing a tool for the automatic translation of programs as developed in task 1.
4. Evaluating the approach on known quantum algorithms

Requirements:

- Basic knowledge in formal methods (like Formal methods lecture)
- Programming experience in Java or C++ (other languages may be possible)
- Basic understanding of linear algebra (Hilbert spaces in particular)

Quellen:

1. Nielsen, Michael A., and Isaac Chuang. "Quantum computation and quantum information." (2002): 558-559.
2. Beckert, Bernhard, Michael Kirsten, Jonas Klamroth, and Mattias Ulbrich. "Modular verification of JML contracts using bounded model checking." In *International Symposium on Leveraging Applications of Formal Methods*, pp. 60-80. Springer, Cham, 2020.