1 Problem Statement

A tool which takes as input a source quantum assembly program and a target hardware and outputs an equivalent program that can be run directly on that hardware.

2 Approach

If the source program is native to the target hardware, i.e. there is a direct compilation path from the source to the target (e.g. qasm to an IBM machine), then that path is taken. Otherwise, the source is compiled through an intermediary language.

3 Classes/Modules

- `static Main`
- `interface Program`
- `interface AssemblyProgram extends Program`
- `interface HardwareProgram extends Program`
- `interface <I, O> Compiler`
- `class IntermediaryProgram implements Program`
- `class QasmProgram implements AssemblyProgram`
- `class QuilProgram implements AssemblyProgram`
- `class SomeSpecificIBMHardwareProgram implements HardwareProgram`
- `class SomeSpecificRigettiHardwareProgram implements HardwareProgram`

4 Logic

The user will use a terminal command to pass in a source program and target hardware. Let's call the source language `I` and the target language `O`.

`Main` will check if a direct compilation path exists. If so, it will use a `<I, O> Compiler` to produce a target program. Otherwise, it will determine the appropriate assembly language `A` use a `<I, IntermediaryProgram> Compiler`, a `<IntermediaryProgram, A> Compiler`, and a `<A, O> Compiler`. `Main` will then output the target program.