Proposal: Phases and Amplitude

1 Synopsis

We will allow phases as roots of unity and amplitudes as an integer associated with each phase.

The goal is to provide a native support for QFT operators as well as an effecient and accurate desription of post-measurement states and outcomes.

1.1 Example

2 Proposal

2.1 Representation

2.1.1 Root of Unity

Notation

$$\omega_N^k = \exp(2\pi i \frac{k}{N})$$

- $\omega_N^0 = 1$
- $\omega_2^1 = -1$
- $\omega_4^1 = i$
- $\omega_4^3 = -i$

2.1.2 Degree of Phases

For efficiency, a hierarchy of phases should be adopted to provide a pay-as-you-go model for phase description. The degree corresponds to the dimension of the underlying matrix.

- zero degree for cases where phases and amplitudes are uninterested
  For example, a simple `Nor` vector where phases can be omitted.
- first degree for cases where one root of unity suffices
  For example, $|-\rangle = \omega_2^0 |0\rangle + \omega_2^1 |1\rangle$.
- second degree for cases using a summation of roots of unity.
  For example,

  $$\sum_{k}^{N-1} \omega_N^{xk} |i\rangle$$

  Note that this is not QFT.

2.1.3 Placement

Phases and amplitudes should be assigned on partition basis. The number of phases object should agree with that of rows per partition.

2.2 Phase Language

- `0`th-degree: Specification without explicit phases are of `0`-degree phases.
- `1`st-degree: `ω(k, N)` for ω_N^k
- `2`nd-degree: `Ω i . (f(i), N)` for

  $$\sum_{i} \omega_N^{f(i)} |\phi\rangle$$

2.3 Data Representation

2.3.1 Zeroth-degree Phase

Zeroth-degree phases, as the name suggests, will not be emitted by the compiler and will be bookkept by the type system. The benefit of this is that there's no verification overhead imposed for a program that doesn't care about phases, e.g,. the GHZ program.

Implementationwise, the modification to the compiler to support the phase will be minimal and incremental.

2.3.2 First-degree Phase

WIP[1]

2.3.3 Second-degree Phase

WIP

2.4 Phase Primitives

2.4.1 Quantum Fourier Transform

$$|x\rangle \mapsto \sum_{k=0}^{N-1} \omega(xk, N) |k\rangle$$

This is a complicated case: the phase itself is of the 1st degree but the outcome state comprises a superposition of state. What makes this even harder is when we apply QFT over a superposition of kets, the norm of doing it on EN states. But, there's a tiny little trick:

$$\sum_{x\in S} |x\rangle \mapsto \sum_{x\in S} \sum_{k=0}^{N-1} \omega(xk, N) |k\rangle = \sum_{k=0}^{N-1} \left(\sum_{x\in S} \omega(xk, N) \right) |k\rangle .$$

Essentially, applying QFT over EN promotes the original zeroth-degree states in superposition into a second-degree state in superposition. We will rely a lot on this equation for reasoning. Note that if the original EN state is of 1st-degree, then the outcome after QFT operation can still be expressed in the second-degree

$$\sum_{(x,i)\in S} \omega(i, N) |x\rangle \mapsto \sum_{(x,i)\in S} \sum_{k=0}^{N-1} \omega(i + xk, N) |k\rangle = \sum_{k=0}^{N-1} \left(\sum_{(x,i)\in S} \omega(i + xk, N) \right) |k\rangle$$

The overhead/difficulty here is how one is going to reason in arithemtics.

In summary, the QFT operator, when applied to kets in superposition, promotes any phase into its second degree representation. If the original repr is of 0th degree, it's straightforward. If it was of 1st degree, we need to match the base of the root of unity through gcm and add them together, which will require a lot of arithmetic reasoning. If we get a 2nd degree, things will be come hard which is equivalent to reasoning about cartisan products.

[1] I'd like to design those two phases while considering the introduction and elemination of those phases, i.e., how phases are transformed into a more expressive one and how expressive phases are contracted/eliminated to extract truth from it.