Proto-Quipper with dynamic lifting

Frank Fu
Joint work with K. Kishida, N.J. Ross and P. Selinger

POPL 2023
Quipper and Proto-Quipper

```
slice2 : !(Qubit -> Qubit -> Bool * Bool)
slice2 a q =
  let (a, q) = CNot a q
  q = U q
  in (dynlift (Meas a), dynlift (Meas q))

bob2 : !(Qubit -> Bool -> Qubit -> Qubit)
bob2 q x y =
  let q = if x then QNot q else q
  q = if y then ZGate q else q
  in q
```

Code

Circuit generation time

Circuit

Circuit execution time

Quantum computer
Dynamic Lifting

Code

Nat, Bool

Circuit

Qubit, Bit

Quantum computer
The two runtimes assumption as categories

- A category of quantum circuits $\mathbf{M}$.
- A category of quantum operations $\mathbf{Q}$.
- An interpretation functor $J : \mathbf{M} \rightarrow \mathbf{Q}$. 
Categorical model for dynamic lifting

- A category $A$ equipped with a monad $T$ such that

\[
\begin{array}{ccc}
M & \longrightarrow & A \\
\downarrow J & & \downarrow \\
Q & \longrightarrow & Kl_T(A)
\end{array}
\]
Categorical model for dynamic lifting

■ A category $A$ equipped with a monad $T$ such that

$$
\begin{array}{ccc}
M & \xrightarrow{} & A \\
\downarrow J & & \downarrow \\
Q & \xleftarrow{} & Kl_T(A)
\end{array}
$$

■ Dynamic lifting is a morphism in $A$ such that

$$
\begin{array}{ccc}
\text{Bool} & \xrightarrow{\text{init}} & \text{Bit} \\
\eta & \xrightarrow{} & \downarrow \text{dynlift} \\
& & T\text{Bool}
\end{array}
$$
Modalities for dynamic lifting

■ $\Gamma \vdash_{\alpha} M : A$, where $\alpha = 0 \mid 1$.

■ Dynamic lifting.

$$\frac{\Gamma \vdash_{\alpha} M : \text{Bit}}{\Gamma \vdash_{0} \text{dynlift} M : \text{Bool}}$$

■ Modality indicates boxability.

$$!_1(S \rightarrow_1 U) \xrightarrow{\text{box}} \text{Circ}(S, U)$$

■ Type system tracks modalities.

$$\frac{\Gamma, x : A \vdash_{\alpha} M : B}{\Gamma \vdash_1 \lambda x.M : A \rightarrow_{\alpha} B}$$

$$\frac{\Gamma_1 \vdash_{\alpha_1} M : A \rightarrow_{\alpha_2} B \quad \Gamma_2 \vdash_{\alpha_3} N : A}{\Gamma_1, \Gamma_2 \vdash_{\alpha_1 \& \alpha_2 \& \alpha_3} MN : B}$$
Categorical semantics

\[\Gamma \vdash_1 M : A \text{ is a map in } A:\]
\[
\llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A \rrbracket
\]

\[\Gamma \vdash_0 M : A \text{ is a map in } Kl_T(A):\]
\[
\llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} T[\llbracket A \rrbracket]
\]
Operational Semantics

- Circuit generation time: $(C, M) \Downarrow (C', V)$
- Circuit execution time: $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i)$

\[
(Q, M) \Downarrow (Q', \ell) \\
(Q, \text{dynlift } M) \Downarrow \text{read}(Q', \ell)
\]

where $\text{read}(Q', \ell) = p_1(Q_1, \text{True}) + p_2(Q_2, \text{False})$. 
Dynamic lifting in practice: quick demo
Main results

- A general categorical model for dynamic lifting.
- A type system uses modality to track dynamic lifting.
- Operational semantics for the two runtimes.
- Type system and operational semantics are sound w.r.t. the categorical semantics.