Online Appendix to MCBeth: A Measurement-based Quantum Programming Language

Aidan Evans, Seun Omonije, and Robert Soulé
Department of Computer Science
Yale University
New Haven, CT, USA
{aidan.evans,seun.omonije,robert.soule}@yale.edu

Robert Rand
Department of Computer Science
University of Chicago
Chicago, IL, USA
rand@uchicago.edu

This document contains the Appendix for the main paper. Section I we discuss the rewriting rules of the measurement calculus in more detail. Section II presents the denotational semantics of MCBeth. Sections III presents a 2-bit Grover's algorithm for MCBeth. Finally, Figure 6 presents several clusters and their corresponding circuits and measurement calculus programs.

I. Rewriting Rules

The rewriting rules from [1] are as follows:

$$\begin{aligned}
X_i^s; E_{ij} &\Rightarrow E_{ij}; Z_j^s; X_i^s \\
X_j^s; E_{ij} &\Rightarrow E_{ij}; Z_i^s; X_j^s \\
Z_i^s; E_{ij} &\Rightarrow E_{ij}; Z_i^s \\
Z_j^s; E_{ij} &\Rightarrow E_{ij}; Z_j^s \\
X_i^r; {}^t[M_i^\alpha]^s &\Rightarrow {}^t[M_i^\alpha]^{s+r} \\
Z_i^r; {}^t[M_i^\alpha]^s &\Rightarrow {}^{r+t}[M_i^\alpha]^s \\
A_{\vec{k}}; E_{ij} &\Rightarrow E_{ij}; A_{\vec{k}} \quad \text{where } A \text{ is not an entanglement} \\
X_i^s; A_{\vec{k}} &\Rightarrow A_{\vec{k}}; X_i^s \quad \text{where } A \text{ is not a correction} \\
Z_i^s; A_{\vec{k}} &\Rightarrow A_{\vec{k}}; Z_i^s \quad \text{where } A \text{ is not a correction}
\end{aligned}$$

where $\vec{k}$ is a set of qubits the command is acting on.

For example, consider again teleportation. We first decomposed

$$J(0)(0,1); J(0)(1,2)$$

into

$$E_{01}; M_0; X_1^{s_0}; E_{12}; M_1; X_2^{s_1}$$

Then, using the rewriting rules, we can rewrite $X_1^{s_0}; E_{12}$ as $E_{12}; Z_2^{s_0}; X_1^{s_0}$. Thus, we now have

$$E_{01}; M_0; E_{12}; Z_2^{s_0}; X_1^{s_0}; M_1; X_2^{s_1}$$

Then, we can rewrite $X_1^{s_0}; M_1$ as $[M_1]^{s_0}$ to get

$$E_{01}; M_0; E_{12}; Z_2^{s_0}; [M_1]^{s_0}; X_2^{s_1}$$

Finally, applying some final rules to commute the commands, we get a standardized program:

$$E_{01}; E_{12}; M_0; [M_1]^{s_0}; Z_2^{s_0}; X_2^{s_1}$$

II. Denotational Semantics

Let P be a program with a computation space of V, inputs I, outputs O, and command sequence $A_1; A_2; ...; A_n$. Let H denote the Hilbert space characterizing the qubits of our computation. The state of our system is characterized by both a Hilbert space of non-measured qubits and an output map containing whether a measured qubit collapsed to equal 0 or 1. A command essentially defines a map from an input state to one or more possible output states. Thus, our semantics will interpret command expressions as relations from states to states.

To begin, for convenience, we treat all of the initial input and preparation commands as one expression. In other words, say that once standardized, commands A_1 through A_i are all the preparation and input commands. For commands $A_1; \ldots; A_i$, the function $\mathcal{C}[\![A_1 \ldots A_i]\!]$ will produce the initialized Hilbert space along with an empty outcome map. Thus,

$$\mathcal{C}[\![A_1 \ldots A_i]\!] = q_0 \otimes q_1 \otimes \ldots \otimes q_n, \varnothing$$

Here, each qubit q_j is in the state it should be after the initialization process so if we had the command $\mathsf{Prep}(2)$, then $q_2 = |+\rangle$. The "$\varnothing$" denotes our currently empty outcome map. Say our outcome map is Γ; note that we'll use $\Gamma[0/j]$ to denote that qubit j maps to 0. Also, we'll use function f as a map of each qubit's number to it's position in the current system; since measurement projects the system down a dimension, the qubit measured is fully removed from H and, therefore, qubit "i" may not be the i^{th} qubit in H anymore.

The remaining denotational semantics for the commands are as displayed in Figure 1.

Note that in the case of measurement we get two branches of our computation: one for each case of how the measured qubit could collapse because, for example, measurement could collapse the qubit to $|+_\alpha\rangle$ or $|-_\alpha\rangle$; therefore, our semantics reflects this by branching the program along one path in which the qubit measured collapses to $|+_\alpha\rangle$ and one for $|-_\alpha\rangle$, reflecting the probabilistic nature of quantum computation.

III. 2-bit Grover's Algorithm

Grover's algorithm allows one to search for a marked element in an unstructured list in $O(\sqrt{n})$ time, where n is

$$\mathcal{C}[\![\texttt{Entangle(i, j)}]\!](H,\Gamma) = E_{f(i),f(j)}H,\Gamma$$
$$\mathcal{C}[\![\texttt{XCorrect(i, ss)}]\!](H,\Gamma) = X_{f(i)}^{\oplus_{s\in ss}\Gamma(s)}H,\Gamma$$
$$\mathcal{C}[\![\texttt{ZCorrect(i, ss)}]\!](H,\Gamma) = Z_{f(i)}^{\oplus_{s\in ss}\Gamma(s)}H,\Gamma$$

$$\mathcal{C}[\![\texttt{Measure(i, }\alpha\texttt{, ss, ts)}]\!](H,\Gamma) = \begin{cases} \langle +_{(-1)^{\oplus_{s\in ss}\Gamma(s)}\alpha+(\oplus_{t\in ts}\Gamma(t))\pi}|_{f(i)} H,\Gamma[0/i] \\ \langle -_{(-1)^{\oplus_{s\in ss}\Gamma(s)}\alpha+(\oplus_{t\in ts}\Gamma(t))\pi}|_{f(i)} H,\Gamma[1/i] \end{cases}$$
$$\mathcal{C}[\![\texttt{J(}\alpha\texttt{, i, j)}]\!](H,\Gamma) = \begin{cases} \langle +_{-\alpha}|_{f(i)} E_{f(i),f(j)}H,\Gamma[0/i] \\ X_{f(j)}\langle -_{-\alpha}|_{f(i)} E_{f(i),f(j)}H,\Gamma[1/i] \end{cases}$$
$$\mathcal{C}[\![\texttt{CZ(i, j)}]\!](H,\Gamma) = E_{f(i),f(j)}H,\Gamma$$
$$\mathcal{C}[\![\texttt{ReadOut(i, }\textit{basis}\texttt{)}]\!](H,\Gamma) = \begin{cases} \langle \psi|_{f(i)} H,\Gamma[0/i] \\ \langle \phi|_{f(i)} H,\Gamma[1/i] \end{cases} \qquad \text{where } \mathcal{S}[\![\textit{basis}]\!] = (|\psi\rangle, |\phi\rangle)$$
$$\mathcal{S}[\![X]\!] = (|+\rangle, |-\rangle)$$
$$\mathcal{S}[\![Y]\!] = (|i\rangle, |-i\rangle)$$
$$\mathcal{S}[\![Z]\!] = (|0\rangle, |1\rangle)$$
$$\mathcal{S}[\![\texttt{FromTuples((a, b), (c, d))}]\!] = \left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix}\right)$$
$$\mathcal{S}[\![\texttt{FromAngle(}\alpha\texttt{)}]\!] = (|+_\alpha\rangle, |-_\alpha\rangle)$$

such that

$$E_{i,j} = M_0 \otimes \ldots \otimes M_{n-1} + N_0 \otimes \cdots \otimes N_{n-1}$$

where n is the total number of qubits in H during this step of the computation,

$$M_i = |0\rangle\langle 0|,\ N_i = |1\rangle\langle 1|,\ N_j = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},\ \text{and all other } M_k, N_k = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$X_i = M_0 \otimes \cdots \otimes M_{n-1} \text{ such that } M_i = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ and all other } M_k = I_2$$
$$Z_i = M_0 \otimes \cdots \otimes M_{n-1} \text{ such that } M_i = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and all other } M_k = I_2$$
$$\langle \psi|_i = M_0 \otimes \cdots \otimes M_{n-1} \text{ such that } M_i = \langle \psi| \text{ and all other } M_k = I_2$$

Fig. 1. Denotational Semantics

the size of the list. It's one of the iconic quantum algorithms used to demonstrate the generality of quantum speedup. Like the Deutsch-Jozsa algorithm, it works by making a call to an oracle function with the input being a superposition of states, $|+\rangle$ to be specific.

Grover's algorithm then works in two parts: (1) using an oracle to mark the desired item in the quantum state and (2) using a diffuser operator to increase this marked item's amplitude and decrease the other items'. Therefore, upon measurement, the item we are searching for will be returned with high-probability.

The oracle and the diffuser work in tandem in a process called *amplitude amplification*, which reflects the state of the system back and forth over a state $|s'\rangle$, which spans both the state of the desired item and the state of the system. In marking the desired item, the oracle performs the initial reflection, and the diffuser applies a negative phase to that result, amplifying the amplitude of the desired item and lowering the amplitudes of all other options. In small search spaces, one iteration of both the oracle and the diffuser is sufficient; however, as the search space scales, multiple calls to this process are necessary to amplify the state of the winner sufficiently.

Once the state of the desired item is properly amplified, measurement of the system will return the state with a significantly higher probability than any other state in the system.

A circuit diagram for a 2-bit Grover's algorithm is shown in

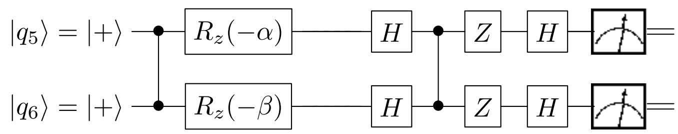

Fig. 2. 2-bit Grover Circuit

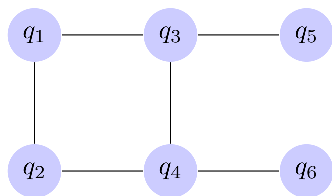

Fig. 3. 2-bit Grover Cluster (6-qubit variation) – corresponds exactly to the circuit in figure 2.

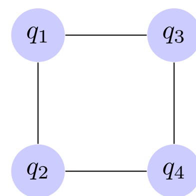

Fig. 4. 2-bit Grover Cluster (4-qubit variation)

```
PrepList([0, 1, 2, 3]);
CZ(0, 1);
J(0, 0, 2);
J(pi, 1, 3);
CZ(2, 3);
ReadOut(2, FromAngle(pi));
ReadOut(3, FromAngle(pi));
```

Fig. 5. 4-qubit variation of Grover's in MCBeth with oracle searching for "10"

Figure 2. The first section of the circuit contains a controlled-Z gate between qubits q_1 and q_2 followed by rotations on each qubit about the Z-axis through an angle of $-\alpha$ and $-\beta$, respectively. This first section represents the oracle and how the oracle is configured is based on the values of α and β. In other words, we set α and β to determine what we are searching for. If, for example, we set $\alpha = 0$ and $\beta = \pi$, then this specifies a search for 1 on our first input qubit and a 0 on our second input qubit. The second section, i.e., the rest of the circuit, represents the reflector.

We present two corresponding possible cluster states for this circuit based on work by [2]: a 6-qubit variation in Figure 3 and a 4-qubit variation in Figure 4 which takes advantage of the readout measurement base like we did in the Deutsch-Jozsa program.

For both variations, the angle of measurement for q_1 and q_2 determines the oracle configuration. An angle of 0 on qubit q_i specifies a search for 1 on bit i and an angle of π specifies a search for 0 on bit i. Then, for the 6-qubit variation, qubits q_3 and q_4 are measured at an angle of 0 and puts the output in qubits q_5 and q_6. Qubits q_5 and q_6 can then be passed as input to another algorithm or they can then be read out in the computational basis to obtain the result. For the 4-qubit variation, we can instead just perform a readout measurement on q_3 and q_4 in the $\{|-\rangle, |+\rangle\}$ basis to obtain the result.

In terms of high-level commands, the 6 qubit variation can be written as follows:

$$CZ(q_1, q_2); J(\alpha)(q_1, q_3); J(\alpha)(q_2, q_4);$$
$$CZ(q_3, q_4); J(\alpha)(q_3, q_5); J(\alpha)(q_4, q_6);$$

followed by readout measurements in the computational basis.

Similarly, or the 4 qubit variation:

$$CZ(q_1, q_2); J(\alpha)(q_1, q_3); J(\alpha)(q_2, q_4); CZ(q_3, q_4);$$

followed by readout measurements in the $\{|-\rangle, |+\rangle\}$ basis. Figure 5 displays the corresponding MCBeth program for the 4-qubit variation.

References

[1] V. Danos *et al.*, "The measurement calculus," *J. ACM*, vol. 54, no. 2, pp. 8–es, Apr 2007.

[2] P. Walther, K. J. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger, "Experimental one-way quantum computing," *Nature*, vol. 434, no. 7030, pp. 169–176, Mar 2005.

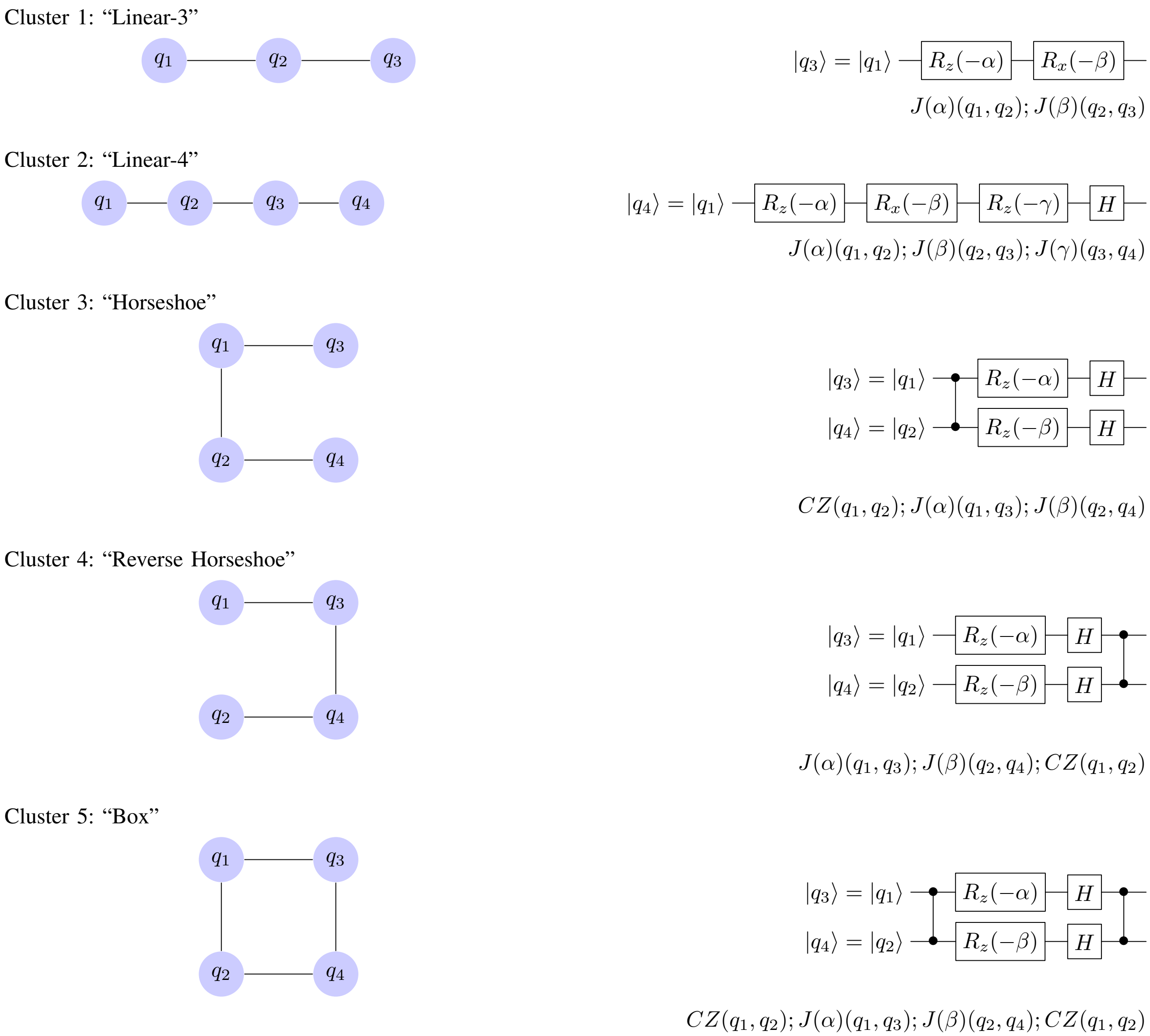

Fig. 6. Some Basic Cluster Conversions. The cluster is shown on the left with the equivalent circuit shown top right and measurement calculus pattern shown bottom right. Based on work by [2].