ZX NOTES

Date: Y.

1 Working plan for Gautham July-September

Overall research question: *How we can use the representation of quantum computations as a MBQC pattern for optimization & compilation tasks?* Every work that fits into this question is useful. The research question splits into two main sub topics:

1. Which graph rewrites are useful on a MBQC pattern?
2. How can we translate MBQC patterns to hardware-adapted instructions?

MBQC patterns can be represented as graph-like diagrams in ZX-calculus.

1.1 Useful graph rewrites

There is no complete rule set for rewriting MBQC patterns so far. Yet, many rules of the complete rule set of standard ZX-diagrams have a similar representation for graph-like diagrams. The bialgebra rule seems to be covered by pivot (resp. local complementation), the copy rule by Z-insertion/Z-deletion, the fusion with again with pivot/ identity removal or neighbor unfusion in the other way.

$\Rightarrow$ Find a complete rule set for graph-like ZX-diagrams.

The most interesting rule for MBQC patterns seems to be local complementation: We can use it to change measurement labels and reduce the number of graph edges in a non trivial way. Optimizing the number of edges with local complementation in general is NP-hard even in the case that all local complementations commute. Yet, if would be interesting to know whether there are restricted graph types for which we can find a sequence of local complementations minimizing the number of edges in polynomial time. One example are graphs locally equivalent to trees. In such cases we can find a sequence of local complementations transforming the graph into a tree which then of course has the minimum number of edges possible. It would be interesting to see whether we can extend the algorithm somehow. Also we could use Pivot instead of local complementation for the same task (although it is restricted in its possibilities)

$\Rightarrow$ Study edge minimization with local complementation.

For an MBQC pattern with Pauli flow, each vertex can be extracted as a Pauli exponential on a quantum circuit, where the underlying graph of the pattern determines the exponential and the Pauli flow determines the order in which we can extract the vertices. We can represent this in a Pauli-dependency DAG. The goal here would be to examine how graph rewrites or changing measurement labels in the Pauli flow affect the exponentials. Yet, determining how the exponentials change is not easy, one can imagine that every operation pushes one or more Pauli strings through the PDDAG updating all Pauli exponentials where the strings anticommute and leaving others unchanged. Still these updates need to be formalized and understood more in order to find algorithms minimizing Pauli exponentials (reducing non-identities in the strings) or getting other desired forms like all Z-terms which are better realizable on neutral atom hardware.

$\Rightarrow$ Study graph and Pauli flow rewrites on PDDAG structure.

1. ZX Introduction

2. Measurements

2.1. Pauli Group.

Definition 1. ***Pauli strings.***

$$P_n = \{\bigotimes_{i=1}^{n} A_i | A_i \in P\}$$

where, P is the Pauli group,

$$P := \alpha\{I, X, Y, Z\}, \alpha \in \{\pm 1, \pm i\}$$

and I, X, Y, Z have matrix representation,

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Note that this is an *irreducible representation*. We can check that X and Z do not share any eigenvectors (and so any invariant 1dim subspaces).

$\mathbb{Q}_8$ is a subgroup of P. P is the smallest subgroup (of $U(2)$) generated by $\langle X, Y, Z \rangle$.

Quotienting P by the center yields the Klein four-group (TODO with other properties and lie algebra lie group).

Note that,

$$SU(2) \cap P_1 = \pm 1\{I\} \cup \pm i\{X, Y, Z\} \leq P_1$$

This subgroup is isomorphic to $\mathbb{Q}_8 = \{\pm 1, \pm i, \pm j, \pm k\}$ via the identification.

$$i \to iX, j \to iY, k \to iZ$$

2.2. Properties.

2.3. Measurements. In an MBQC we need to specify measurement operations for each non-output. We do this by assigning *measurement planes* for each qubit,

Any general measurement for a single qubit is specified by an axis on the Bloch sphere. Convention is to restrict these axes to a plane of the Bloch sphere – XY, YZ, or XZ. The axes selects two states – $|\eta\rangle$ and $|\eta'\rangle$ which is diametrically opposite on the sphere, then we form,

$$\Pi = |\eta\rangle\langle\eta| + |\eta'\rangle\langle\eta'|$$

and make a projective measurement.

We can write these states and axes explicitly. The Bloch sphere is parametrised as (θ, φ) – azimuthal and polar. We choose our axes by fixing one of the angles,

$$\theta = \pi/2 \ (XY), \varphi = 0 \ (YZ) \text{ or } \varphi = \pi/2 \ (XZ)$$

$$|+_{XY}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha}|1\rangle) \qquad |-_{XY}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\alpha}|1\rangle)$$

$$|+_{XZ}\rangle = \cos\frac{\alpha}{2}|0\rangle + \sin\frac{\alpha}{2}|1\rangle \qquad |-_{XZ}\rangle = \sin\frac{\alpha}{2}|0\rangle - \cos\frac{\alpha}{2}|1\rangle$$

$$|+_{YZ}\rangle = \cos\frac{\alpha}{2}|0\rangle + i\sin\frac{\alpha}{2}|1\rangle \qquad |-_{YZ}\rangle = \sin\frac{\alpha}{2}|0\rangle - i\cos\frac{\alpha}{2}|1\rangle$$

Note that any point (pure state) on the boundary of the Bloch sphere can be written as,

$$(\theta, \varphi) \mapsto \cos\frac{\varphi}{2}|0\rangle + e^{i\theta}\sin\frac{\varphi}{2}|1\rangle$$

The measurement axis coincide with X, Y or Z corresponds to $\alpha = a\pi$ with $a \in \{0, 1\}$ [1]

[1] more exactly $\alpha = \frac{2a}{\pi} \mod 4, a \in \{0, 1, 2, 3\}$ to pick out both axes in each plane.

After picking an axis, we construct our measurement as a projector,

$$\Pi_{AB,\alpha} = |+_{AB}(\alpha)\rangle\langle +_{AB}(\alpha)| - |-_{AB}(\alpha)\rangle\langle -_{AB}(\alpha)|$$

Typically in MBQC, our "desired outcome" is the $+1$ eigenvalue collapse to the $|+\rangle$. This is usually denoted as outcome 0, and the undesired $|-\rangle$ collapse as outcome 1.

Depending on the plane of measurement, we can apply a pauli X, Y or Z to correct it. Z changes the relative phase, X swaps 0 and 1 and $ZX = Y$ does both.

$$Z|-_{XY}\rangle = |+_{XY}\rangle,\ Y|-_{XZ}\rangle = |+_{XZ}\rangle \text{ and } X|-_{YZ}\rangle = |+_{YZ}\rangle$$

2.4. **Stabilisers.**

Definition 2. *The* ***Clifford group*** *is the* normaliser *of the Pauli group P_n in $U(2^n)$,*

$$C_n := \{g \in SU(2^n) | gP_ng^{-1} = P_n\}$$

A ***Clifford gate*** *is an element of C_n.*

fact: the Clifford group on n-qubits is generated by Hadamard, Phase (i) and CNOT gates.

Theorem 3. ***(Gottesmann-Knill).*** *Any circuit involving only Clifford gates and Pauli measurements is (polynomial time) easily simulatable.*

So, we want to try to minimize the number of non-Clifford gates in our circuit.

fact: C_n is not a universal gate-set. The T gate cannot be finitely generated.

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Note that the Clifford group is not finite because if g normalises P_n then so does $e^{i\phi}g$. We can disregard global phases and just consider, $C_n/U(1)$.

3. MBQC

4. References

1. Pauli and Stabilisers – Watrous TQI
2. MBQC –
3. PDDAG –

4.1. **Stabilizer–Based Feed-Forward Corrections in MBQC.** Let $G = (V, E)$ be our open graph with inputs $I \subset V$ and outputs $O \subset V$, and let $|G\rangle$ be the associated graph-state on V:

$$\mathcal{S}_G \;=\; \Big\langle K_u = X_u \prod_{v \in N(u)} Z_v : u \in V \Big\rangle, \qquad K_u|G\rangle = +|G\rangle \quad \forall\, u.$$

We process measurements in an order consistent with a flow f on (G, I, O). **Initialization.**

- For each non-output qubit $u \in V \setminus O$, choose:
  - A measurement plane $\lambda(u) \in \{XY, XZ, YZ\}$.
  - A *base angle* $\phi_u \in [0, 2\pi)$.
- Initialize two counters on every qubit $v \in V$,
$$c_X(v) \;=\; 0, \qquad c_Z(v) \;=\; 0.$$
  These record how many π–shifts have been accumulated for the X– and Z–components of v's basis.

Online Loop. Iterate over $u \in V \setminus O$ in flow order:

(1) *Compute the adaptive angle.* If $\lambda(u) = XY$, set
$$\alpha_u \;=\; \phi_u \;+\; \pi\, c_X(u).$$
If $\lambda(u) = XZ$, use $\pi\, c_Z(u)$ in place of $\pi\, c_X(u)$, and similarly for YZ by combining both.

(2) *Measure qubit u.* Project onto
$$\Pi_u^{(s_u)} = \frac{1 + (-1)^{s_u}\, P_u(\alpha_u)}{2}, \quad P_u(\alpha_u) = \cos\alpha_u\, X_u + \sin\alpha_u\, Y_u$$
let $s_u \in \{0, 1\}$ be the outcome bit (0 for "+", 1 for "–").

(3) *Syndrome-based feed-forward.* For each *future* qubit $v \succ u$:
$$c_X(v) \;\longmapsto\; c_X(v) + s_u \quad \text{if } [P_v,\, P_u(\alpha_u)] \neq 0 \text{ in the } X\text{–component,}$$
and
$$c_Z(v) \;\longmapsto\; c_Z(v) + s_u \quad \text{if } [P_v,\, P_u(\alpha_u)] \neq 0 \text{ in the } Z\text{–component.}$$
Equivalently, v lies in the *anticommuting neighbourhood* of u under the stabilizers K_v.

Final Correction on Outputs.

$$\text{On each } o \in O, \text{ apply } X^{\,c_X(o)} Z^{\,c_Z(o)} \quad \text{(or flip the classical output bits accordingly).}$$

Why this works:

- Each measurement at u "flips" those stabilizers K_v which anticommute with $P_u(\alpha_u)$, exactly encoding the syndrome bit s_u.
- Adding $\pi\, s_u$ to the angle of each affected future v cancels the corresponding $(-1)^{s_u}$ byproduct.
- Because the flow order guarantees causality (u only affects $v \succ u$), all byproducts are absorbed before reaching the outputs, yielding a deterministic overall map.

5. Pauli Flow

6. PDDAG