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.

⇒ 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)

⇒ 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.

⇒ Study graph and Pauli flow rewrites on PDDAG structure.

1. Measurements

1.1. Pauli Group.

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

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

where, P is the Pauli group,

$$P := \alpha\{I, X, Y, Z\}, \alpha \in \{\pm 1, \pm i\} \tag{2}$$

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

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3}$$

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,

$$SU(2) \cap P_1 = \pm 1\{I\} \cup \pm i\{X, Y, Z\} \leq P_1 \tag{4}$$

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

$$i \to iX, j \to iY, k \to iZ \tag{5}$$

1.2. Properties. [TODO]

1.3. 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 are diametrically opposite on the sphere, then we form,

$$\Pi = |\eta\rangle\langle\eta| + |\eta'\rangle\langle\eta'| \tag{6}$$

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,

$$\theta = \pi/2 \ (XY), \varphi = 0 \ (YZ) \text{ or } \varphi = \pi/2 \ (XZ) \tag{7}$$

$$\begin{aligned} |+_{XY}\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha}|1\rangle) & |-_{XY}\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - e^{i\alpha}|1\rangle) \\ |+_{XZ}\rangle &= \cos\frac{\alpha}{2}|0\rangle + \sin\frac{\alpha}{2}|1\rangle & |-_{XZ}\rangle &= \sin\frac{\alpha}{2}|0\rangle - \cos\frac{\alpha}{2}|1\rangle \\ |+_{YZ}\rangle &= \cos\frac{\alpha}{2}|0\rangle + i\sin\frac{\alpha}{2}|1\rangle & |-_{YZ}\rangle &= \sin\frac{\alpha}{2}|0\rangle - i\cos\frac{\alpha}{2}|1\rangle \end{aligned}$$

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

$$(\theta, \varphi) \mapsto \cos\frac{\varphi}{2}|0\rangle + e^{i\theta}\sin\frac{\varphi}{2}|1\rangle \tag{8}$$

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,

$$\Pi_{\bullet\circ,\alpha} = |+_{\bullet\circ}(\alpha)\rangle\langle +_{\bullet\circ}(\alpha)| - |-_{\bullet\circ}(\alpha)\rangle\langle -_{\bullet\circ}(\alpha)| \tag{9}$$

It is useful to construct the measured Hermitian operators (with eigenbasis $|+\rangle, |-\rangle$) corresponding to each planar measurement.

$$\begin{aligned} XY &\mapsto \cos\alpha X + \sin\alpha Y =: M_{XY,\alpha} \\ XZ &\mapsto \sin\alpha X + \cos\alpha Z =: M_{XZ,\alpha} \\ YZ &\mapsto \sin\alpha Y + \cos\alpha Z =: M_{YZ,\alpha} \end{aligned}$$

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.

$$Z|-_{XY}\rangle = |+_{XY}\rangle,\ Y|-_{XZ}\rangle = |+_{XZ}\rangle \text{ and } X|-_{YZ}\rangle = |+_{YZ}\rangle \tag{10}$$

2. Stabilisers

The idea is to specify a state (uniquely) via the generators of its stabiliser subgroup. "Errors" become changes to generators.

2.1. **Clifford group.**

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

$$C_n := \{g \in SU(2^n) | gP_ng^{-1} = P_n\} \tag{11}$$

*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 initial state $|0\rangle^{\otimes n}$, 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.

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{12}$$

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 $C_n/U(1)$.

2.2. **Stabilisers and States.**

Definition 4. *The **subspace stabilised** by a subgroup $H \leq P_n$ is,*

$$V_H := \{|\psi\rangle \in (\mathbb{C}^2)^{\otimes n} | h|\psi\rangle = |\psi\rangle \forall h \in H\} \tag{13}$$

Proposition 5. *For all $g \in U(2^n)$ and $H \leq P_n$, $V_{gHg^{-1}} = gV_H$.*

Note that if $-\mathbb{1} \in H$, then $-\mathbb{1}|\psi\rangle = |\psi\rangle \implies |\psi\rangle = 0$. This must be excluded for the stabiliser space to be non-trivial.

Proposition 6. *If $H \leq P_n$ and* $\dim(V_H) > 0$*, then H is abelian and $-\mathbb{1} \notin H$.*

Proof. $g_1, g_2 \in P_n$ either commute or anticommute. If they anticommute then $g_1 g_2|\psi\rangle = -g_2 g_1|\psi\rangle \implies |\psi\rangle = 0$ □

Definition 7. *A set $S \leq H$ is* ***independent*** *if for all $g \in S$,*

$$\langle S \setminus \{g\}\rangle \neq \langle S\rangle \tag{14}$$

And finally, the theorem below allows us to specify a unique state by specifying n commuting Pauli strings. Let us denote V_S to be the subspace stabilised by a generating set $\langle S\rangle$.

Theorem 8. *If $S = \{g_1, \dots, g_l\}$ is indepdent, pairwise commutative such that $-\mathbb{1} \notin S$, then $\dim(V_S) = 2^{n-l}$.*

There is a surjective group homomorphism, $r : P_n \to \mathbb{F}_2^{2n}$ with $\ker(r) = \{\pm 1, \pm i\mathbb{1}\}$ via the map,

$$r(X_i) = e_i \text{ and } r(Z_i) = e_{i+n} \tag{15}$$

we can keep track of X_i, Y_i via this map as a $2 \times 2n$ matrix called the check matrix.

For example, the bell state $(|00\rangle + |11\rangle)$ is the stabiliser state of,

$$\langle X_1 X_2, Z_1 Z_2\rangle \tag{16}$$

with associated check matrix,

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \tag{17}$$

The stabiliser tableaux is an extension of the check matrix of the state.

Since we can specify states via its stabiliser (group), we can also track measurements. First, note that for any $g \in P_n$, the projectors P_+, P_- can be written as,

$$P_+ = \frac{1}{2}(I + g) \text{ and } P_- = \frac{1}{2}(I - g) \tag{18}$$

with probabiltiies,

$$p(+) = \frac{1}{2}(1 + \langle\psi|\tilde{g}|\psi\rangle) \text{ and } p(-) = \frac{1}{2}(1 - \langle\psi|\tilde{g}|\psi\rangle) \tag{19}$$

We want to measure some observable $\tilde{g} \in P_n$.

It is easy to check that any $g \in P_n$ either commutes or anticommutes with $\tilde{g}$. Consider a stabiliser V_S generated by $S = \langle g_1, \dots, g_n\rangle$. There are two cases,

(i) $\tilde{g}$ commutes with all stabilisers.

Then, for any g, $g(\tilde{g}|\psi\rangle) = \tilde{g}|\psi\rangle$. So, $\tilde{g}|\psi\rangle$ is a common $+1$ eigenvector of all stabilisers. Since V_S is one-dimensional, $\tilde{g}|\psi\rangle \propto |\psi\rangle$, but because it has eigenvalues ± 1, $\tilde{g}|\psi\rangle = \pm|\psi\rangle$. So, one of $\pm\tilde{g}$ is a stabiliser.

The outcome is deterministic $+$ or $-$ if $+\tilde{g} \in S$ or $-\tilde{g} \in S$. We do not need to update the stabilisers,

$$S = \langle g_1, \dots, g_n\rangle \tag{20}$$

(ii) $\tilde{g}$ anticommutes with some stabilisers.

wlog, we can assume that $\tilde{g}$ anticommutes with g_1 and commutes with all other g_i by picking g_1 such that $\{\tilde{g}, g_1\} = 0$ and for other i, $g_i \mapsto g_1 g_i =: g_i'$,

$$g g_i' = g(g_1 g_i) = -g_1(g g_i) = -g_1(-g_i g) = g_i' g \tag{21}$$

note that this is just a change of choice of generators and does not change S.

In this case, the measurement is *not* deterministic because,

$$p(+) = \frac{1}{2}(1 + \langle\psi|\tilde{g}|\psi\rangle) = \frac{1}{2}(1 + \langle\psi|\tilde{g}g_1|\psi\rangle) = \frac{1}{2}(1 - \langle\psi|\tilde{g}|\psi\rangle) = p(-) = \frac{1}{2} \tag{22}$$

In this case, depending on the measurement outcome we update the stabiliser (after measurement) to,

$$S = \langle \pm\tilde{g}, g_2, \dots, g_n\rangle \tag{23}$$

The GHZ state for example cannot be described by a stabiliser because $p(Z_1 = +) = \frac{1}{3}$.

Note that all of this assumes that we are making a *pauli measurement.* In an MBQC, before we make the X_u local measurement, we must update the tableaux to the above form – anticommuting with one (neighbour) vertex and commuting with all others as in 21.

This update for a X-measurement is the exact same as local complementation on the measured vertex (prove later).

3. MBQC

MBQC leverages entanglement (teleportation) instead of unitary gates to create a one-way model of quantum computation.

Definition 9. *A* ***measurement pattern*** *consists of an n-qubit register V with distinguished sets $I, O \subseteq V$ of inputs and outputs. Additionally there is a sequence of the following operations:*

i. Preperations – intialising all qubits $i \notin I$ in the $|+\rangle$ state.

ii. Entangling – applying a CZ_{ij} to qubit i as control and qubit j as target for (chosen) pairs $(i, j) \in V \times V$.

iii. Destructive measurements – which project qubits $i \notin O$ onto orthonormal basis $\{|+_{\lambda,\alpha}, |-_{\lambda}, \alpha\rangle\rangle\}$ as described in 1.3.

iv. Corrections – conditionally applying X or Z gates onto qubits $i \in V$.

The *graph* (state) for a measurement pattern is determined by the set of tuples $(i, j) \in V \times V$ such that we apply CZ_{ij}. Along with specifying measurement axes and angles for non-outputs $\bar{O} := V \implies$, we get a labelled open graph.

Definition 10. *A* ***labelled open graph*** *is a tuple $\Gamma = (G, I, O, \lambda)$ where $G = (V, E)$ is a graph, $I, O \subseteq V$ are (input and output) subsets of V and $\lambda : \bar{O} \to \{XY, YZ, XZ\} \times [0, 2\pi)$.*

Every measurjement yields an outcome 0 or 1. In total there are $2^{|\bar{O}|}$ possible outcomes – these are called *branches* of the measurement pattern.

After corrections we can sometimes ensure that every branch yields the same output. When this is possible, the measurement pattern is called **deterministic**. This is formalised via *flow.*

The **linear map** associated with a deterministic MBQC then looks like,

$$\underbrace{\left(\prod_{i\in\bar{O}} \langle +_{\lambda(i),\alpha(i)}|\right)}_{\text{(det) measurement}} \underbrace{\left(\prod_{i\sim j} CZ_{ij}\right)}_{\text{entangling}} \underbrace{\left(\prod_{i\in\bar{I}} |+\rangle_i\right)}_{\text{preperation}} \tag{24}$$

This acts on the input state $\bigotimes_{i\in I} |\psi_i\rangle$. The *resource state* used is,

$$\left(\prod_{i\sim j} CZ_{ij}\right) \left(\prod_{i\in\bar{I}} |+\rangle_i\right) |\psi\rangle_I \tag{25}$$

3.1. **Corrections.** Note that we cannot simply apply corrections by applying a Z or X gate on an incorrect outcome (typically a hardware/cost limitation on MBQC). Let us construct the stabiliser of the graph state.

First note that $|+\rangle_u = X_u|+\rangle_u$ and the identity,

$$CZ_{uv}X_u = X_uZ_vCZ_{uv} \tag{26}$$

Then, for any $w \in \bar{I}$,

$$\left(\prod_{u\sim v} CZ_{u,v}\right)\left(\prod_{u\in\bar{I}} |+\rangle_u\right) = \left(\prod_{u\sim v} CZ_{u,v}\right) X_w \left(\prod_{u\in\bar{I}} |+\rangle_u\right) \tag{27}$$

$$= \left(\prod_{v\in N_G(w)} Z_v\right) X_w \left(\prod_{u\sim v} CZ_{u,v}\right) \tag{28}$$

where we use the identity above when CZ involves vertex w, and otherwise it commutes with X_w. Also, the Z string commutes with X_w.

So, we see that,

$$\left[\left(\prod_{v\in N_G(w)} Z_v\right) X_w\right]\left(\prod_{u\sim v} |+\rangle_u\right) = \left(\prod_{u\sim v} |+\rangle_u\right) \tag{29}$$

For each vertex we get a stabiliser string of Z on the neighbours and X on the vertex.

Equivalently, we obtain the following operator equivalence that will allow us to perform corrections,

$$X_w\left(\prod_{u\sim v} |+\rangle_u\right) = \left(\prod_{v\in N_G(w)} Z_v\right)\left(\prod_{u\sim v} |+\rangle_u\right) \tag{30}$$

The above case is easy – to correct by applying a local X on vertex w can be achieved by applying a Z to all neighbours of w.

We can also take the product of stabilisers indexed (defined) by many vertices. To corrct by applying a local Z, we can take the stabiliser of vertices in in its neighbourhood.

Generally, we look for a stabiliser of the graph state,

$$S = Q_v \otimes P_{rest} \tag{31}$$

where Q_v is the local correction operator on vertex v. This is dependent (only) on the choice of plane and angle. The string of paulis P_{rest} is called an *extraction string.* Simple algebraic manipulation of a chosen stabiliser is the generator for corrections.

We want P_{rest} to be supported on the future (unmeasured) qubits. This is possible when we have a *flow* condition on the graph.

3.2. **Flow.** The simplest case is causal flow, we assume that all vertices are measured in the XY basis.

Definition 11. ***(causal flow).***
Given a labelled open graph $\Gamma = (G, I, O, \lambda)$ such that $\lambda(v) = XY$ for all $v \in \bar{O}$, a causal flow is a tuple $(f, \prec)$ where $f : \bar{O} \to \bar{I}$ and $\prec$ is strict partial order on V satisfying,

i. $v \sim f(v)$

ii. $v \prec f(v)$

iii. $\forall u \in N_G(f(v))$*,* $u = v$ *or* $v \prec u$

The partial ordering $\prec$ gives us the order in which to perform measurements.

Here, an error on any $v \in \bar{O}$ can be corrected via applying Z_v. Consider the stabiliser of vertex $f(v)$, rearranging we get,

$$\prod_{w \in N_G(f(v) \setminus \{v\})} Z_w X_{f(v)} = Z_v \tag{32}$$

Allowing measurements in all three planes and products of vertex stabilisers gives us generalised flow.

The following algebraic relation is very useful, let the stabiliser be specified by a set $g(v) \subseteq \bar{I}$,

$$\begin{aligned}
\prod_{u \in g(v)} \left(\prod_{w \in N_G(u)} Z_w \right) X_u &= \left(\prod_{u \in g(v)} \prod_{w \in N_G(u)} Z_w \right) \left(\prod_{u \in g(v)} X_u \right) && (33)\\
&= \left(\prod_{u \in \mathrm{odd(g(v))}} Z_u \right) \left(\prod_{u \in g(v)} X_u \right) && (34)\\
&= \left(\prod_{u \in \mathrm{odd}(g(v)) \setminus g(v)} Z_u \right) \left(\prod_{u \in g(v) \cap \mathrm{odd}(g(v))} Y_u \right) \left(\prod_{u \in g(v) \setminus \mathrm{odd}(g(v))} X_u \right) && (35)
\end{aligned}$$

where, in the second line we use that $Z_w^2 = I$, and that only Z on the odd neighbourhood of $g(v)$ survive.

We then write the expression as a disjoint product – either,

i. $u \in \mathrm{odd}(g(v))$ and $u \notin g(v)$ – apply Z

ii. $u \in g(v)$ and $u \in \mathrm{odd}(g(v))$ – apply Y

iii. $u \in g(v)$ and $u \notin \mathrm{odd}(g(v))$ – apply X

Finally, based on whether we require a X_v, Y_v or Z_v correction, we choose $g(v)$ appropriately such that v belongs to one of the corresponding sets. We search for subsets $g(v)$ and the ordering $\prec$ simultaneously, not separately.

Definition 12. ***(generalised flow).***
Given a labelled open graph $\Gamma = (G, I, O, \lambda)$ such that $\lambda(v) \in \{XY, XZ, YZ\}$ for all $v \in \bar{O}$, a generalised flow *or* glow *for Γ is a tuple $(g, \prec)$ where $g : \bar{O} \to \mathcal{P}(\bar{I})$ and $\prec$ is a strict partial order over V satisfying for all $v \in \bar{O}$,*

i. for all $u \in g(v)$, $v \neq u \implies v \prec u$

ii. for all $u \in \mathrm{odd}(g(v))$, $v \neq u \implies v \prec u$

iii. $\lambda(v) = XY \implies v \in \mathrm{odd}(g(v)) \setminus g(v)$

iv. $\lambda(v) = XZ \implies v \in g(v) \cap \mathrm{odd}(g(v))$

v. $\lambda(v) = YZ \implies v \in g(v) \setminus \mathrm{odd}(g(v))$

Conditions i, ii ensure that correction sets are in the *future*, and iii - v follow from the stabiliser algebra.

We can generalise this by noting that if some measurements are pauli, $\lambda(u) \in \{X, Y, Z\}$, then u can be in the correcting set of some v in the future ($u \prec v$) as long as correcting v induces the same pauli $\lambda(u)$ on u. This is described as a *free correction in the past.* Note that the ± sign must still be tracked, but doesn't require the application of any operator.

Definition 13. ***(Pauli flow).***
Given a labelled open graph $\Gamma = (G, I, O, \lambda)$, a Pauli flow *for Γ is a tuple $(p, \prec)$ where $p : \bar{O} \to \mathcal{P}(\bar{I})$ and $\prec$ is a strict partial order over V satisfying for all $v \in \bar{O}$,*

$[\prec .X]$ *for all $u \in p(v)$, $v \neq u$ and $\lambda(u) \notin \{X, Y\} \implies v \prec u$.*

$[\prec .Z]$ *for all $u \in \mathrm{odd}(p(v))$, $v \neq u$ and $\lambda(u) \notin \{Y, Z\} \implies v \prec u$.*

$[\prec .Y]$ *for all $v \not\prec u$, $v \neq u$ and $\lambda(u) = Y \implies u \notin p(v) \triangle \mathrm{odd}(p(v))$*

$[\lambda.XY]$ $\lambda(v) = XY \implies v \in \mathrm{odd}(p(v))$ *and* $v \notin p(v)$

$[\lambda.XZ]$ $\lambda(v) = XZ \implies v \in p(v)$ *and* $v \in \mathrm{odd}(p(v))$

$[\lambda.YZ]$ $\lambda(v) = YZ \implies v \in p(v)$ *and* $v \notin \mathrm{odd}(p(v))$

$[\lambda.X]$ $\lambda(v) = X \implies v \in \mathrm{odd}(p(v))$

$[\lambda.Z]$ $\lambda(v) = Z \implies v \in p(v)$

$[\lambda.Y]$ $\lambda(v) = Y \implies v \in p(v) \triangle \mathrm{odd}(p(v))$

where, $\triangle$ denotes the symmetric difference of sets.

The first three conditions are on the ordering $\prec$ similar to gflow, with the additional exceptional case for planar measurements in the past. The other six conditions are identical to gflow, with the last three explicitly written for the planar case.

We can describe the stabiliser algebra for the correcting set $p(v)$ as,

$$S(p(v)) = \prod_{w \in p(v)} X_w \prod_{w \in \mathrm{odd}(p(v))} Z_w \tag{36}$$

The local operator at some vertex $u \in p(v)$ is,

$$P_u = X^{1_{u \in p(v)}} Z^{1_{u \in \mathrm{odd}(p(v))}} \tag{37}$$

where, $1_{x \in A}$ denotes the characteristicsfunction – 1 if $x \in A$ and 0 if $x \notin A$.

The third axiom $[\prec .Y]$ describes the $(0, 0)$ corresponding to a local I and $(1, 1)$ corresponding to a local Y case. Note that writing $u \notin p(v) \cap \mathrm{odd}(p(v))$ would not allow the trivial identity case.

In Pauli flow, past vertices with pauli measurements are allowed to be in future correcting sets as long as the induced local operator commutes.

Focussed flow requires that this is *necessarily* the case for every correction regardless of ordering. That is, the induced $P_{v \to u} \in \{\mathrm{I}, \lambda(u)\}$. Here,

$$P_{v \to u} := X^a Z^b, \quad a = 1_{u \in p(v)}, \ b = 1_{u \in \mathrm{odd}(p(v))} \tag{38}$$

whether $v \prec u$ or $u \prec v$.

Definition 14. ***(focussed pauli flow).***
a pauli flow $(p, \prec)$ is focussed if for all $u, v \in \bar{O}$ and $u \neq v$,

$$P_{v \to u} \in \{I, \lambda(u)\}$$

There is an amibiguity here with treating non-planar measurements, so we identify measurement planes with the pauli at angle 0.

A bit more generally,

Definition 15. ***(focussed sets)*** *Given a labelled open graph Γ, a set $\tilde{p} \subseteq \bar{I}$ is focussed over $S \subseteq \bar{O}$ if,*

$[FX]$ for all $v \in S \cap \tilde{p}$, $\lambda(v) \in \{XY, X, Y\}$

$[FZ]$ for all $v \in S \cap \mathrm{odd}(\tilde{p})$, $\lambda(v) \in \{XZ, YZ, Y, Z\}$

$[FY]$ for all $v \in S$, $\lambda(v) = Y \implies (v \in \tilde{p} \iff v \in \mathrm{odd}(\tilde{p}))$

$\tilde{p}$ is a focussed set *for Γ if it is focussed over $\bar{O}$.*

A pauli flow $(p, \prec)$ is a ***focussed pauli flow*** *if for all $v \in \bar{O}$, $p(v)$ is focussed over $\bar{O} \setminus \{v\}$. We can check that in this case the definition of a focussed set coincides with the condition in definition 14.*

4. PDDAG AND REWRITE RULES

4.1. **Extraction strings.** Recall 31, we want to use focussed pauli flow, write every measurement as a rotation + pauli, then finally push the rotations to just the outputs. This requires identifying a stabiliser $S_v = Q_v \otimes P_{rest}$ for each vertex where Q_v is the local correction on vertex v, so orthogonal to $\lambda(v)$.

To be precise, $P_{rest} \equiv P_{abs} \otimes P_{outs}$ can consist of any operations on the outputs, and also local paulis that are absorbed by other vertices [2] 15 – the fact that we can always find such a stabiliser S_v is guaranteed by focussed pauli flow. The following rotation lemma computes the 'extraction strings' on the outputs.

Lemma 16. ***(rotation lemma),***
Let Q, S be commuting pauli strings, and let S be a stabiliser of $|G\rangle$ [3]. Then,

$$e^{-i\frac{\theta}{2}Q}|G\rangle = e^{-i\frac{\theta}{2}QS}|G\rangle$$

To compute the extraction strings, we use the rotation lemma and set $Q = Q_v$ and $S = S_v$ – then, in the product QS we are left with $\mathbb{1} \otimes P_{rest}$.

The only problem that remains is that if $u \in p(v), u \succ v$, and $\lambda(u)$ is planar, then $P_{rest}^{v \to u}$ might not be absorbed by $\lambda(u)$. To resolve this, we process vertices in reverse order of flow (i.e, increasing measurement depth order) and at each step convert the measurement labels into paulis while pushing the rotations to the output.

4.2. **PDDAG.**

Definition 17. ***(extraction strings)*** *Let (Γ, α) be a measurement pattern with focussed pauli flow $(p, \prec)$. Let $v \in \bar{O}$. Then, a pauli string P is a Q-extraction string ($Q \in \{X, Y, Z\}$) for v if $Q_v P_O$ is a stabiliser of,*

$$\left(\prod_{\substack{u \in \bar{O} \\ u \succ v \\ \lambda(u) \neq \{X,Y,Z\}}} \langle +_{\lambda(u),0}| \right) \left(\prod_{\substack{u \in \bar{O} \setminus \{v\} \\ \lambda(u) \in \{X,Y,Z\}}} \langle +_{\lambda(u),0}| \right) \left(\prod_{u \sim w} CZ_{u,w} \right) \left(\prod_{u \in \bar{I}} |+\rangle_u \right)$$

The first projector (on the right) are for pauli measurements, which can be performed whenever (not part of the PDDAG), and the second projector is for all vertices that have already been extracted and written to the outputs.

[2] pauli flow will only guarantee that paulis are absorbed by past vertices, not future

[3] or more generally, of any linear map C

5. Rewrites for Circuit Simplification

Let Γ be a measurement pattern with focussed pauli flow $(p, \prec)$ and $v \in \bar{O}$. Define the following set,

$$A(v) := \{p(v) \cup \mathrm{odd}(p(v))\} \tag{39}$$

Then,

Definition 18. *The* ***pauli-weight*** *of vertex $v \in \bar{O}$ is given by,*

$$\omega(v) := |A(v) \cap O|$$

The ***pauli-weight*** *of the pattern is then,*

$$\omega(\Gamma) := \sum_{\substack{v \in \bar{O} \\ \lambda(v) \notin \{X,Y,Z\}}} \omega(v)$$

Note that when local complementation is applied to the graph, we must update the flow,

$$(p, \prec) \xrightarrow{\text{loc compl.}} (p', \prec) \tag{40}$$

Then, our problem reduces to applying local complementation such that,

$$\underbrace{\sum_{v \in \bar{O}} |\left(p'(v) \cap \mathrm{odd}(p'(v))\right) \cap O|}_{\omega(\Gamma')} \leq \underbrace{\sum_{v \in \bar{O}} |\left(p(v)\right) \cap \mathrm{odd}(p(v)) \cap O|}_{\omega(\Gamma)} \tag{41}$$

Concretely, local complementation updates all correction sets (and thusly odd neighbourhoods) according to *Lemma D.14* from Simmons' paper. For all $v \in \bar{O}$, local complementation about a vertex $u \in \bar{O}$ yields,

$$p'(v) := \begin{cases} p(v) \triangle \{u\} & v \in \mathrm{odd}(p(v)) \\ p(v) & u \notin \mathrm{odd}(p(v)) \end{cases} \tag{42}$$

Note that this is the new pauli flow (not focussed!). The focussed condition is a bit more complicated to account for various cases [*Lemma D.15*].

5.1. Matrix formalism.

An idea is to compute the sets of interest with the following matrix ingredients, and see how local complementation changes it.

1. Adjacency matrix A

2. Matrix of correction sets P, where for a fixed vertex v, $p(v)$ is the indicator column vector $\in \{0,1\}^V$

3. The product AP (modulo 2) will yield the matrix of odd neighbourhoods.

Then finally,

i. Define $W := P \vee AP$ (entrywise OR sum)

ii. Extract the rows corresponding to the outputs O

iii. Delete the columns corresponding to pauli vertices and obtain the reduced W_ω

iv. $\omega(\Gamma) := \sum_{i,j} W_{\omega ij}$

We can also compute the new correction sets after local complementation via $P' = P \oplus AP$ (modulo 2) [TODO and CHECK]

6. Local Complementation

We want to come up with some sort of strategy for how to reduce pauli-weight by applying Local complementation on the graph. We will try to do this by obtaining some analytic expression for pauli-weight after LC based on Appendix D from the paper.

First, we demonstrate that it is only refocussing that changes the pauli-weight. The updated correction sets after LC on the graph (to obey Pauli flow) leaves the weights unchanged, but the focussing can change the weights.

We can show that pauli-weight is unchanged until focussing, following Lemma D.14. In the raw LC update, only a single vertex is toggled (or not) in the new correction set. So the odd neighbourhood is straightforward to compute.

Some identities we will use:

Proposition 19. *(Linearity of* odd*)*

$$\mathrm{odd}(A\triangle B) = \mathrm{odd}(A)\triangle\mathrm{odd}(B)$$

From D.13,

Proposition 20.

$$\mathrm{odd}_{G\star u}(A) = \begin{cases}\mathrm{odd}_G(A)\triangle(N_G(u)\cap A) & u\notin \mathrm{odd}_G(A)\\ \mathrm{odd}_G(A)\triangle(N_G(u)\setminus A) & u\in \mathrm{odd}_G A)\end{cases}$$

Using this we want to compute either $\mathrm{odd}_{G\star u}(p(v)\setminus w)$ or $\mathrm{odd}_{G\star u}(p(v)\cup w)$.

Also, since LC is an involution – $(G\star u)\star u = G$ this makes it sufficient to show one direction.

Proposition 21. *Let X be a set and $A : X \to \mathcal{P}(X)$ be an assignment. Let f be an involution on X $(f(f(x)) = x$ for all $x\in X)$. If for all $x\in X$, $A(x)\subseteq A(f(x))$, then, $A(x) = A(f(x))$ for all $x\in X$.*

In our case, the involution is defined as follows, for any $u\in V$ we have the involution,

$$f_u(G, p(v)) = (G\star u, p'(v))$$

This is an involution because $\star^2 = \mathrm{id}$ and $((p(v)\triangle u)\triangle u) = p(v)$.

Theorem 22. *Let (Γ, G, α) be a pattern and after LC at $u\in\bar{O}$, $(\Gamma', G\star u, \alpha)$ be the updated pattern. Then,*

$$A(v) := p(v)\cup\mathrm{odd}_G(p(v)) = p'(v)\cup\mathrm{odd}_G(p(v)) =: A'(v)$$

where,

$$p'(v) = \begin{cases}p(v)\triangle\{u\} & u\in\mathrm{odd}_G(p(v))\\ p(v) & u\notin\mathrm{odd}_G(p(v))\end{cases}$$

Proof. (by cases). We only need to consider $v\in\bar{O}$ such that $u\notin\mathrm{odd}_G(p(v))$. Let $w\in V$ be arbitrary,

<u>case 1. $w = u$</u>

case (i): $u\notin\mathrm{odd}_G(p(v))$. So we have $p(v) = p'(v)$.

Let $u\in A(v) \implies u\in p(v) \implies u\in p'(v)\subseteq A'(v)$.

Let $u\in A'(v)$. From D.13,

$$\mathrm{odd}_{G\star u}(p(v)) = \mathrm{odd}_G(p(v))\triangle(N_G(u))\cap p(v)$$

But u cannot be in the RHS, so we must have that $u\in p'(v) \implies u\in p(v)\subseteq A(v)$.

case (ii): $u\in\mathrm{odd}_G(p(v))$. So $p(v)\triangle p'(v) = \{u\}$. Let $u\in A(v)$.

subcase(a). $u \in p(v) \implies p'(v) = p(v) \setminus \{u\}$. So,

$$\text{odd}_{G\star u}(p'(v)) = \text{odd}_{G\star u}(p(v)) \triangle odd_{G\star u}(\{u\})$$

But since $u \in \text{odd}_G(p(v))$,

$$\text{odd}_{G\star u}(p(v)) = \text{odd}_G(p(v)) \triangle (N_G(u) \setminus p(v))$$

and $\text{odd}_{G\star u}(\{u\}) = N_{G\star u}(\{u\}) = N_G(\{u\})$.

Putting it together with associativity of $\triangle$,

$$\text{odd}_{G\star u}(p'(v)) = \text{odd}_G(p(v)) \triangle (N_G(u) \cap p(v))$$

So, $u \in \text{odd}_{G\star u}(p'(v)) \subseteq A'(v)$.

subcase(b). $u \notin p(v) \implies u \in p'(v) \subseteq A'(v)$.

The converse, to show that $u \in A'(v) \implies u \in A(v)$ follows from the same arguments or 21.

case 2. $w \neq u$ and $w \notin N_G(u)$

Local complementation does not change the adjacency of w, so, $N_{G\star u}(w) = N_G(w)$. If $w \in p(v)$ then $w \in p'(v)$ and we are done. Finally,

$$\text{odd}_{G\star u}(p(v)) = \text{odd}_G(p(v)) \triangle (N_G(u) \setminus \{u\})$$

So $w \in \text{odd}_{G\star u}(p(v))$ if $w \in \text{odd}_G(p(v))$

case 3. $w \neq u$ and $w \in N_G(u)$

Let us assume that $w \in A(v)$.

case (i): $u \in odd_G(p(v))$.

Here we repeat the calculation from 1. case (ii) (using D.13 and linearity, associativity of odd),

$$\text{odd}_{G\star u}(p'(v)) = \text{odd}_G(p(v)) \triangle (N_G(u) \cap p(v))$$

So, $w \in \text{odd}_G(p(v)) \implies w \in \text{odd}_{G\star u}(p'(v))$

case (ii): $u \notin \text{odd}_G(p(v))$

Then, $w \in p(v) \implies w \in p'(v)$ from 21.

Therefore, we have shown that $w \in A(v) \implies w \in A'(v)$ and from 21 the other direction follows. Before focussing the flow, local complementation leaves the weight invariant. □

The change in pauli-weight comes purely from taking the symmetric set diference again with the correction sets for the following candidates,

$$W(v) = \left\{ w \in A(v) \cap (N_G(u) \cup \{u\}) \middle| w \succ v \text{ and } \begin{cases} \text{if } w = u \text{ then } \lambda(u) \notin \{X, Y, Z\} \\ \text{if } w \in N_G(u) \text{ then } \lambda(w) = XY \end{cases} \right\}$$

So, we should choose vertices u for LC such that they are not planar measurements, or their neighbours are XY measurements.

We can perform a similar computation as above for the updated weight after focussing.

To distinguish, let us denote the update correction sets before focussing (obeying pauli flow) as $\tilde{p}$. The pauli weight for a vertex v after local complementation about some vertex u is,

$$p'(v) \cup \text{odd}p'(v) = (\tilde{p}(v) \cup \text{odd}(\tilde{p}(v))) \triangle \mathop{\triangle}_{w \in W(v)} \big(p'(w) \cup \text{odd}(p'(w))\big)$$

$$p'(v) \cup \text{odd}p'(v) = \left(\tilde{p}(v) \triangle \mathop{\triangle}_{w \in W(v)} p'(w) \right) \cup \left(\text{odd}(\tilde{p}(v)) \triangle \mathop{\triangle}_{w \in W(v)} \text{odd}(p'(w)) \right)$$

intersected with O.

Note that this is not simply $A'(v) = \tilde{A}(v) \triangle_w A'(w)$

w must initially satisfy two properties:

(i) $w \succ v$

(ii) $w \in N_G(u)$ or $w = u$

An element vanishes on the RHS if it appears a total even number of times in the $|W(v) + 1|$ sets.

To summarise, to reduce $\omega(v)$, we need to find a suitable LC vertex u. For this to have any effect on the weight, we require:

(i) $u \in A(v)$

Then, consider vertices either $w \in N_G(u)$ or $w = u$. We must take the difference of these sets for which $w \succ v$, either,

(ii) $w \in N_G(u) \implies \lambda(w) = XY$

(iii) $w = u \implies \lambda(u) \notin \{X, Y, Z\}$

So, by looking at different $\lambda(u)$ [TODO]

6.1. **Calculation of correction sets via chains.** We seek to derive a non-recursive method (closed form) of computing correction sets after local complementation.

As a simplifying assumption, let us ignore the LC vertex u itself. Consider the candidate set W for a vertex x. This is the future subset of $N_G(u)$ in $A(x)$ with XY labels.

Working with matrices over $\mathbb{F}_2$, the $\triangle$ operation is simply addition modulo 2. The familiar update rule is,

$$p'(x) = p(x) + \bigoplus_{w \in W} p'(w)$$

If the vertices in W were totally ordered: $0 \succ 1 \succ 2 \succ \ldots \succ n$ then,

$$p'(0) = p(0), p'(1) = p(1) + p(0), p'(2) = p(2) + p(1) + p(0) + p(0) = p(2) + p(1)$$

The general pattern is $p'(k) = p(k) + p(k-1)$.

When there is no total ordering it is a bit more complicated. For example in a 'diamond',

$$x \prec a \prec y \text{ and } x \prec b \prec y$$

with $a, b, y \in W$.

Calculating similarly, we will find that $p'(x) = p(x) + p(a) + p(b) + p(y)$.

The general rule is as follows.

Proposition 23. *For a vertex $x \in \bar{O}$ and $y \in W$, $p'(x)$ will include $p(y)$ in the sum iff there are an odd number of strictly $\prec$-increasing sequences starting at x and ending at y.*

$$p'(x) = p(x) \oplus \bigoplus_{y \in W} \mathcal{C}(x, y) p(y) \tag{43}$$

where, $\mathcal{C}(x, y) \in \{0, 1\}$ is the number of strictly $\prec$-increasing sequences from x to y modulo 2.

(i) Total order: Number of sequences between any two non-adjacent vertices is a power of 2. Which is why only the adjacent vertex appers as $p'(k) = p(k) + p(k-1)$.

(ii) Diamond: Sequences from x to y are three in total: $x \prec a \prec y$, $x \prec b \prec y$, $x \prec y$.

General strategy to compute the new correction set for $x \in \bar{O}$ after LC:

(i) Identify the candidate set W_x and arrange it as a poset.

(ii) For each vertex $w \in W_x$, calculate the number of $\prec$-increasing paths from x to w and keep track of the parity. Then compute $p'(x)$ as in 43.

6.2. **Partial $\triangle$-Invariance of correction sets.** Consider a single local complementation,

$$0 \xrightarrow{LC_u} 1$$

With $G \mapsto G \star u,\ p_0 \mapsto p_1$. If $|I| = |O|$, this is an involution.

Note that:

(i) LC does not change $\prec$ on G.

(ii) For any $A_0 \subset N_G(u)$, $A_0 = A_1$ because $N_{G\star u}(u) = N_G(u)$

So, for any $x \in G$,

$$p_1(x) = p_0(x) \triangle_{w \in W_0} p_1(w)$$

where, $W_0 \subset N_G(u)$ with $\lambda(w) = XY$, $w \succ x$ and $w \in p_0(x) \cup \mathrm{odd}(p_0(x))$

Applying LC at u again we restore the original pattern,

$$p_0(x) = p_1(x) \triangle_{w \in W_1} p_0(w)$$

here, $W_1 \subset N_G(u)$ with $\lambda(w) = XY$, $w \succ x$ and $w \in p_1(x) \cup \mathrm{odd}(p_1(x))$

Taking the symmetric difference of both expressions,

$$p_0(x) \triangle p_1(x) = p_0(x) \triangle p_1(x) \triangle_{w \in W_1} p_0(w) \triangle_{w \in W_0} p_1(w)$$

Therefore,

$$\triangle_{w \in W_1} p_0(w) = \triangle_{w \in W_0} p_1(w)$$

Again, the $w \in A(x)$ filter for the candidate set complicates the expression.

7. Pivot

We want to show that when $|I| = |O|$, a pivot leaves extraction strings invariant.

Conjecture. *Let $\Gamma = (G, p, \prec)$ be a pattern with focussed pauli flow and $|I| = |O|$. Then, after a pivot on any edge (u, v), $u, v \in \bar{I}$, the updated pattern $\Gamma' = (G', p', \prec)$ leaves the extraction strings invariant. That is, for all $x \in G$,*

$$p(x) \cap O = p'(x) \cap O \text{ and } \mathrm{odd}(p(x)) \cap O = \mathrm{odd}(p'(x)) \cap O$$

Below is a collection of ideas/strategies/attempts so far.

7.1. **Observations.** Consider the pivot about an edge (u, v) We are interested in the effect of this operation on $x \in \bar{O}$.

A pivot involves a sequence of 3 local complementations, so there are 4 total states of the graph/measurement pattern:

$$G^0 \xrightarrow{LC_u} G^1 \xrightarrow{LC_v} G^2 \xrightarrow{LC_u} G^3$$

Similarly, the correction set of $p(x)$ changes as,

$$p^0(x) \xrightarrow{LC_u} p^1(x) \xrightarrow{LC_v} p^2(x) \xrightarrow{LC_u} p^3(x)$$

Our goal is to show that $p^0(x) \cap O = p^3(x) \cap O$.

Some observations.

(1) Pivot is symmetric in (u, v). i.e, $G \wedge uv = G \wedge vu$.

(2) Excluding the LC vertex u, the label updates for other vertices leave the candidates invariant. That is, the vertex sets $\{w | \lambda(w) = XY\}$ and $\{w | \lambda(w) \notin \{X, Y, Z\}\}$ are invariant.

exceptional case: If we consider a pivot $G \wedge uv$, then if $\lambda_0(v) = YZ$, it will contribute since $YZ \mapsto XZ \mapsto XY$.

(3) LC (and hence pivot) leaves $\prec$ invariant.

(4) Define $A = N_G(u) \setminus \{v\}, B = N_G(v) \setminus \{u\}$, and A_{XY}, B_{XY} be corresponding subsets with $\lambda = XY$. Then, we have a sequence of three corresponding correction sets for each update, $W_0 \to W_1 \to W_2$.

It follows from tracking the pivot that, $W_0 = A_{XY}, W_1 = B_{XY} \triangle A_{XY}$ and $W_2 = B_{XY} \triangle A_{XY} \triangle A_{XY} = B_{XY}$. Notice that $W_1 = W_0 \triangle W_2$. This is a naturally symmetric characterisation of the neighbourhood sets.

Issue: This does not account for the fact that there is another filter, namely that we must select $w \in W_i$ only if they belong in $p_i(x) U \mathrm{odd}(p_i(x))$. That breaks the above identity.

(5) When $|I| \neq |O|$, I notice that the conjecture breaks specifically for vertices $x \in I$. The proof must rely on the fact that $|I| = |O|$. Collecting relevant results from the paper:

   (a) *Lemma B.2,B.7:* $\triangle$ of focussed sets is focussed. When $|I| = |O|$ the focussed set over $\bar{O} \setminus \{v\}$ is unique.

   (b) *Lemma B.9:* $2^{|O|-|I|}$ distinct focussed sets. This might be what is breaking the conjecture on the inputs.

(6) *Definition 4.3:* because $p(x)$ is focussed over the pivot neighbourhood, then from the definition of a focussed set it follows that $\lambda(w) = XY$ and $w \in p(x) \cup \mathrm{odd}(p(x)) \implies w \in p(x)$. So it is sufficient to filter candidates by intersecting with $p(x)$.

(7) The proof must also involve taking an intersection with the outputs. For this, it seems most natural/direct to work with extraction strings and update rules directly.

(8) *Lemma B.7, Duncan:* Tells us how to compute $\text{odd}(p_3(x))$ in terms of membership conditions of u, v in $\text{odd}_{G_0}(p_0(x))$. Involves toggling $\triangle$ with $N_{G_0}(u)$ and $N_{G_0}(v)$.

7.2. **Proof and Simplification Strategies.**

(1) Strong induction in reverse $\prec$-order seems powerful.

(2) Equalities among $\triangle_{w \in W_i} p_i(w_i)$ for different i can be obtained by equating $\wedge uv = \wedge vu$ and the fact that LC is an involution.

(3) Deriving some combinatorial results to use with the chains approach 43.

7.3. **Matrix Framework for Pivot Invariance.** Let (Γ, λ) be a measurement pattern with a focussed pauli flow $(p, \prec)$. Let $x \in \bar{O}$ and consider a pivot across an edge (u, v) such that u, v do not appear as candidates.

Define $A = N_G(u) \setminus \{v\}, B = N_G(v) \setminus \{u\}$, and A_{XY}, B_{XY} be corresponding candidate subsets with $\lambda = XY$. Then, we have a sequence of states of the graph $G_0 \to G_1 \to G_2 \to G_3$ with three corresponding correction (candidate) sets for each update, $W_0 \to W_1 \to W_2$.

It follows from the pivot that, $W_0 = A_{XY}, W_1 = B_{XY} \triangle A_{XY}$ and $W_2 = B_{XY} \triangle A_{XY} \triangle A_{XY} = B_{XY}$. Notice that $W_1 = W_0 \triangle W_2$.

For any set $W \subseteq \bar{O}$, let use define a matrix T_W over $\mathbb{F}_2$,

$$T_W(x, y) = \begin{cases} C(x, y) & \text{if } y \in W \\ 0 & \text{otherwise} \end{cases}$$

Here, $C(x, y)$ is the number of chains, i.e, the number of strictly $\prec$-increasing sequences from x to y.

$$p'(x) = p(x) \oplus \bigoplus_{y \in W} C(x, y) p(y)$$

This detects whether $p(y)$ appears in the $\triangle$-update for $p(x)$.

Now, we can compute how the correction sets for any vertex x transform, $p_0 \to p_1 \to p_2 \to p_3$,

$$p_1 = (I + T_{W_0})p_0, \ p_2 = (I + T_{W_1})p_1, \ p_3 = (I + T_{W_2})p_2$$

So,

$$p_3 = (I + T_{W_2})(I + T_{W_1})(I + T_{W_0})p_0 =: Pp_0$$

Now substituting $T_{W_1} = T_{W_0} + T_{W_2}$ and simplifying, we get,

$$P = I + T_{W_0}^2 + T_{W_2}^2 + T_{W_2}T_{W_0} + T_{W_2}^2 T_{W_0} + T_{W_2}T_{W_0}^2 =: I + D$$

then,

$$p_3(x) = p_0(x) \bigoplus_{y \in \bar{O}} D(y)p(y)$$

The correction set is invariant, $p_3 = p_0$ if all the quadratic terms disappear. Also, from symmetry $u \wedge v = v \wedge u$ so,

$$T_{W_2}T_{W_0} + T_{W_2}^2 T_{W_0} + T_{W_2}T_{W_0}^2 = T_{W_0}T_{W_2} + T_{W_0}^2 T_{W_2} + T_{W_0}T_{W_2}^2$$

Let us now compute $\mathrm{Odd}_{G \wedge uv}(p_3(x))$ by applying *Lemma D.13* recursively,

$$\mathrm{odd}_{G \star u}(S) = \begin{cases} \mathrm{odd}_G(S) \triangle (N_G(u) \cap S) & u \notin \mathrm{odd}_G(S) \\ \mathrm{odd}_G(S) \triangle (N_G(u) \setminus S) & u \in \mathrm{odd}_G S) \end{cases}$$

Making the simplifying assumption that u, v never appear in $\mathrm{odd}_G(p_3(x))$,

$$\mathrm{odd}_{G \wedge uv}(p_3(x)) = \mathrm{odd}_G(p_3(x)) \triangle (N_G(u) \setminus S) \triangle (N_{G \star u}(v) \setminus S) \triangle (N_{G \star u \star v}(u) \setminus S)$$

This simplifies because we have $A, A \triangle B, B$ to,

$$\mathrm{odd}_{G \wedge uv}(p_3(x)) = \mathrm{odd}_G(p_3(x))$$

Use Lemma B.7 from Duncan for other cases.