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.

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

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

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