March update

finding detection regions

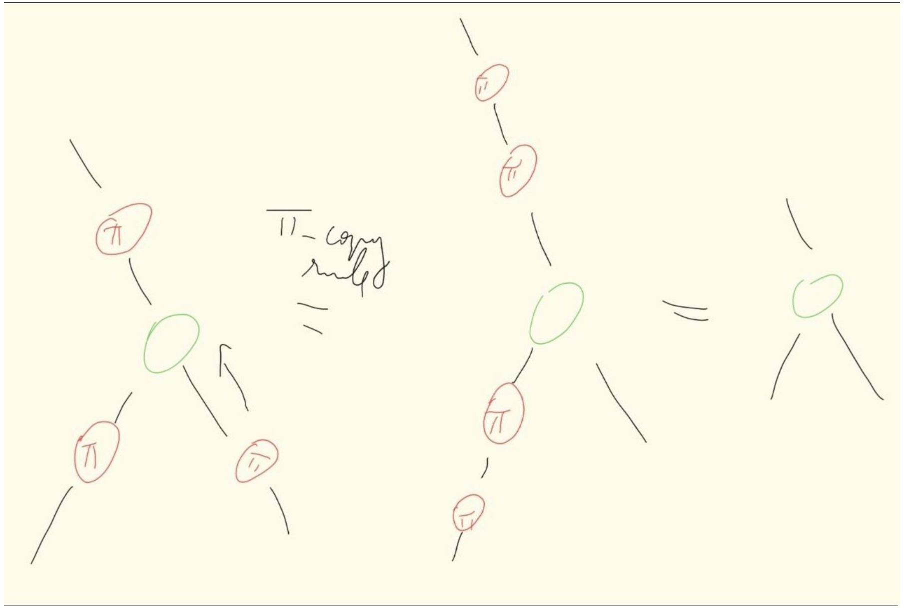
The ZX calculus

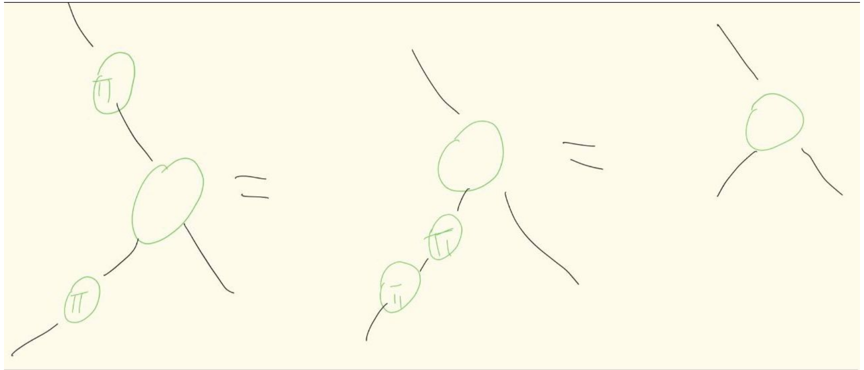
- Spanned by these 7 rules…
-

Pauliwebs

- We can “fire” each spider in ways that leave the diagram unchanged

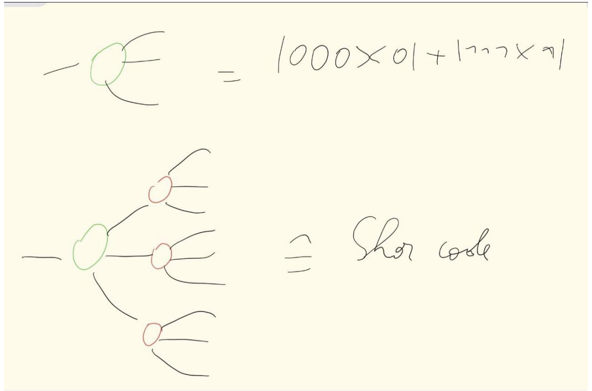
- All or none in opposite-color, or an even number in same-color

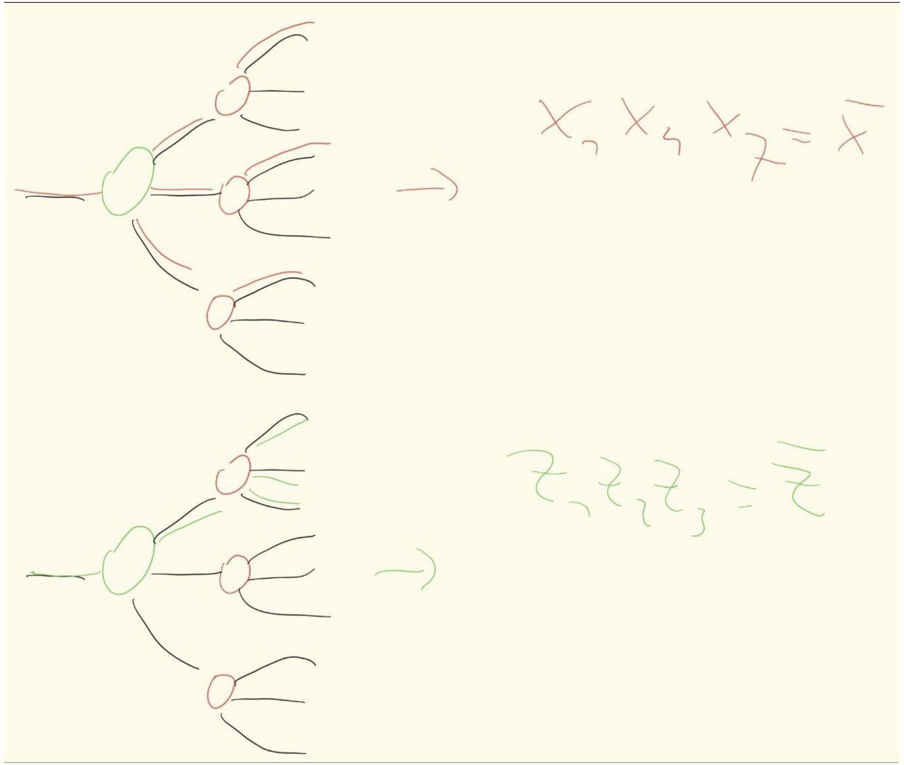




Pauliwebs

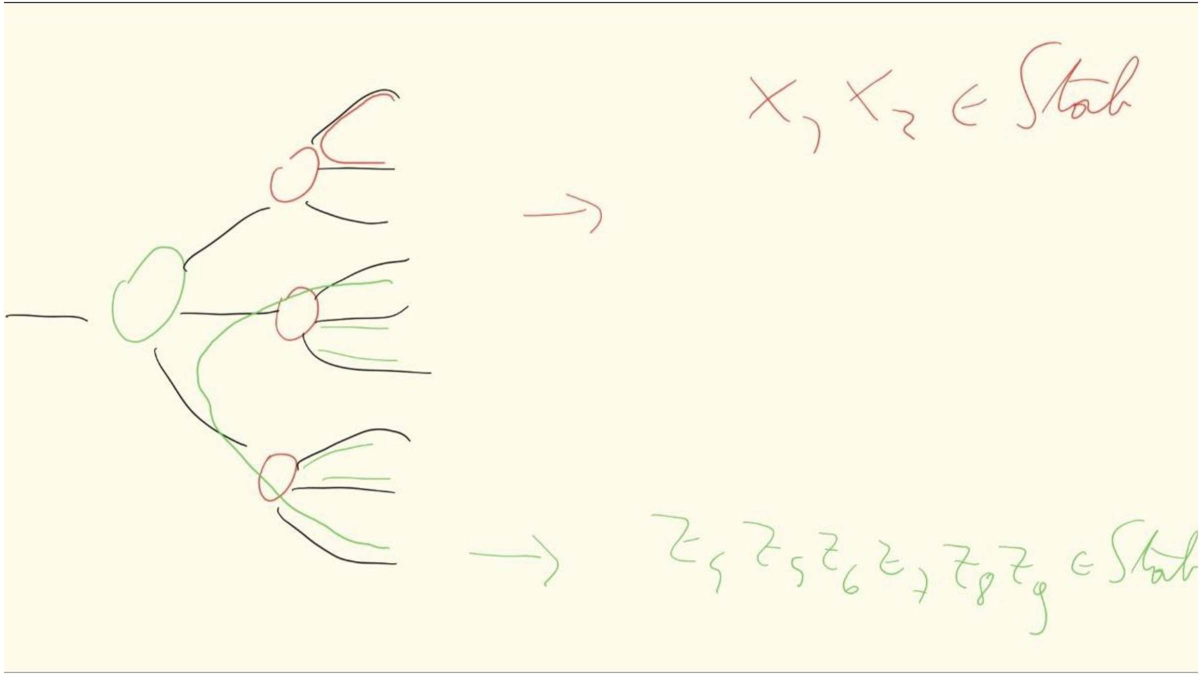
- In larger diagrams, these can form connected webs
- If a web contains input and output edges, it can tell us how to propagate an operator through diagram





Pauliwebs

- We can also use Pauliwebs to find stabilizers of an encoding map

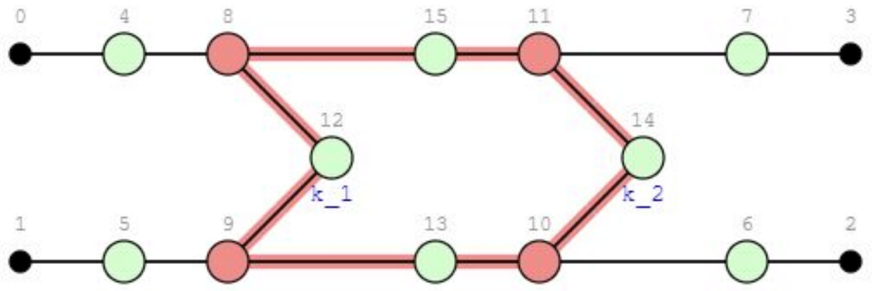


But this is a very static/time-local perspective!

Instead: detecting regions!

- An internal pauliweb to a diagram forms a **detection region**

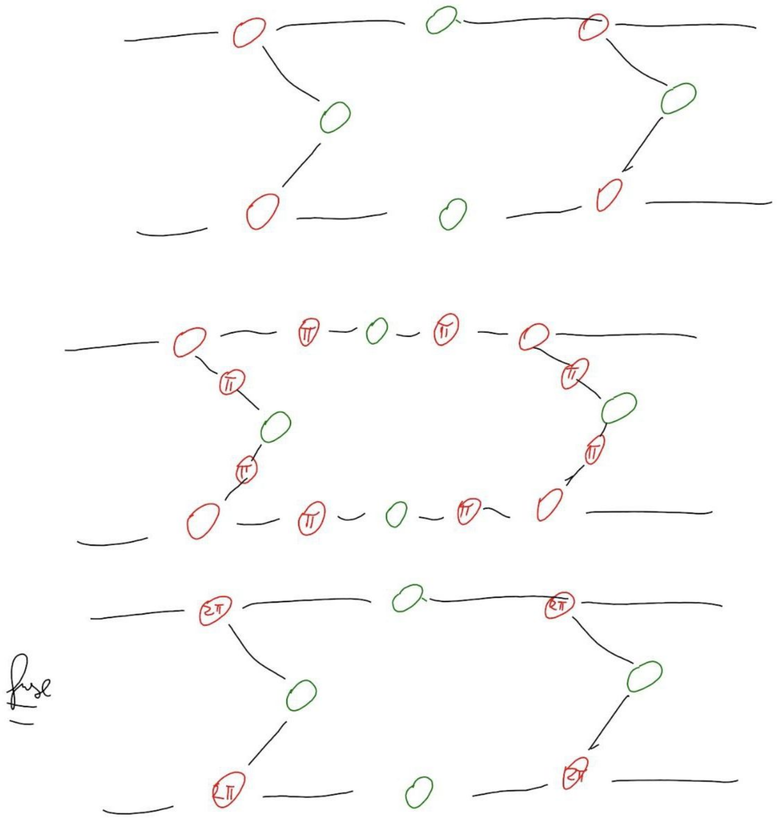
- The parity of incident measurements to that web will be even in the absence of errors, odd otherwise



Example

Consider the 2 XX measurements:

In the no-error case, firing spiders changes nothing

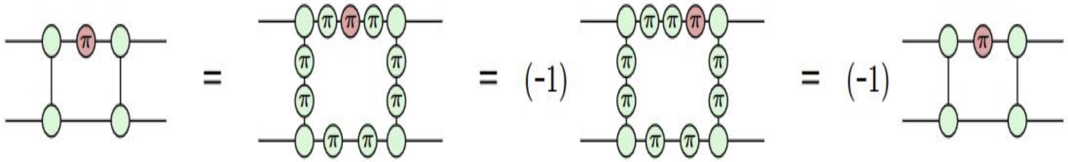


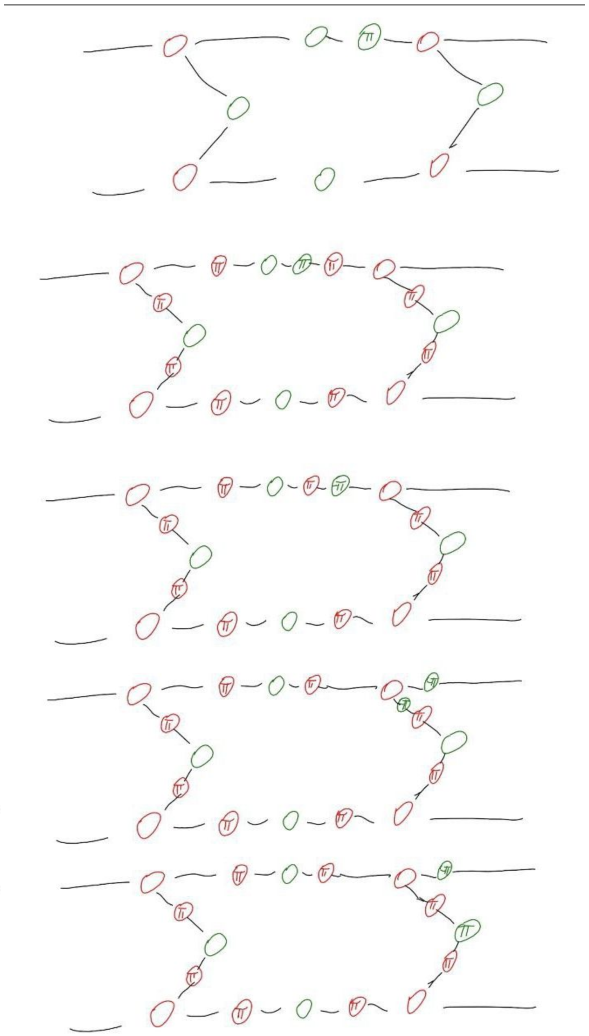
Example

Consider the 2 XX measurements:

In the Z error case, it propagates to one of the measurements, resulting in a changed outcome parity

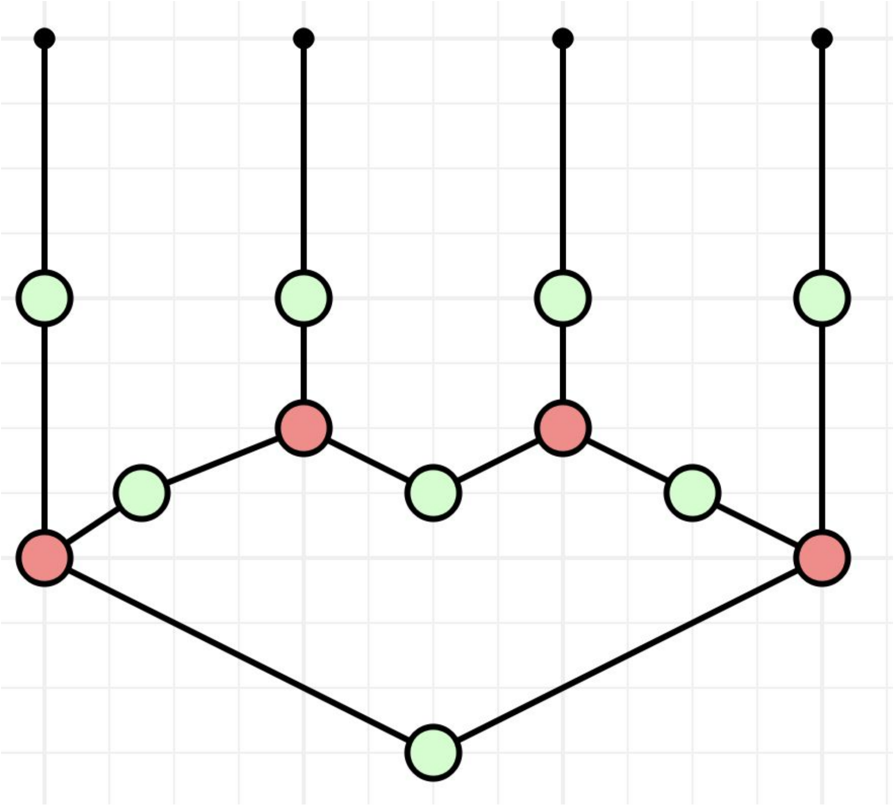
This means our web anticommutes with this error, it forms a **detector** for it





How to find a pauliweb

- Take red-green form of Choi state of the diagram (here for the two XX measurements)
- Assign binary variable to each internal node for firing
- Ensure that sum of firing neighbors is even $$\sum_{x:(q,x)\in E} x = 0 mod 2$$



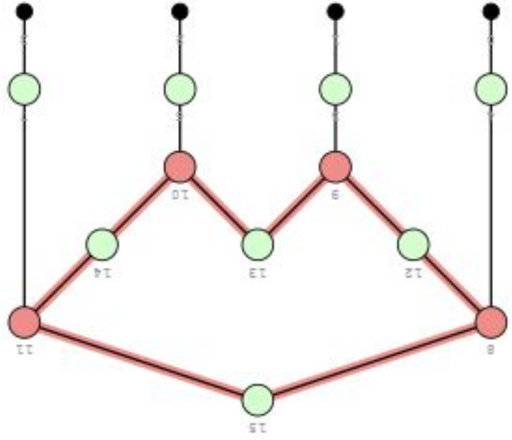
How to quickly calculated valid firing vectors

The previous condition is equivalent to finding the Kernel of a matrix of the form:

$$M_D := \left(\begin{matrix} I_n \\ \emptyset \end{matrix} \middle| \, N \right)$$

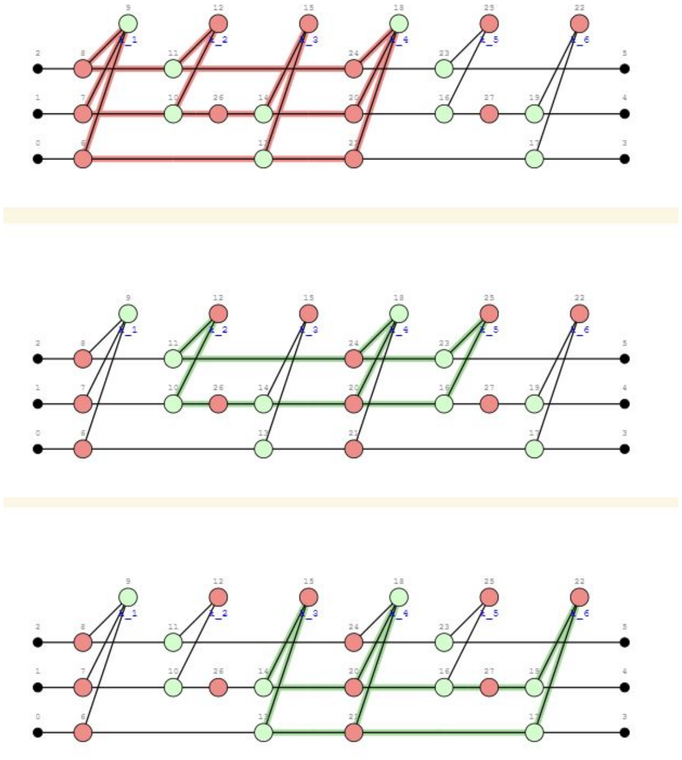
Where N is the adjacency matrix

We can further restrict to no outputs firing by adding rows with single entries to that matrix, since Kernel won't contain vectors with entries there



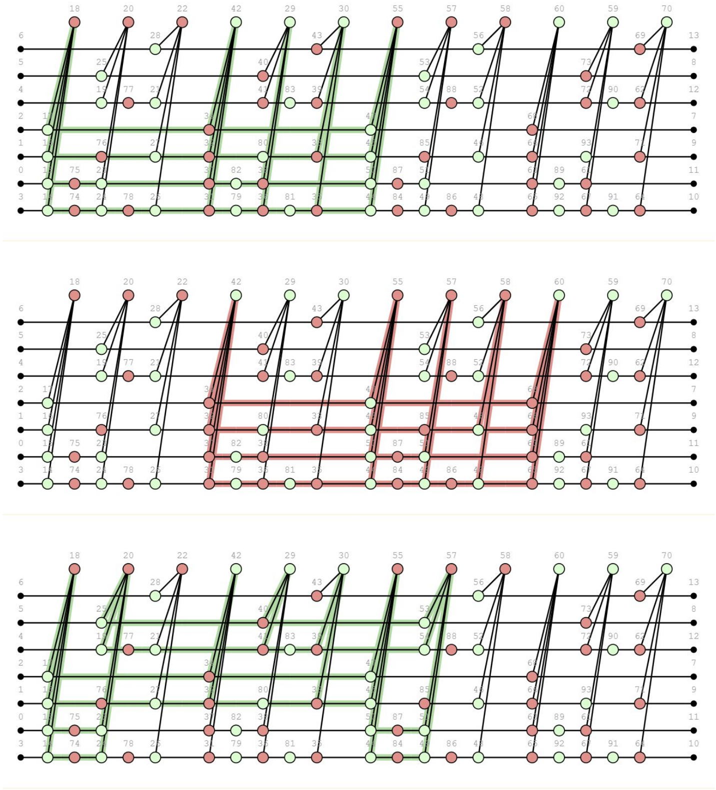
Detection regions on a star graph state (<XXX,ZZI,IZZ>)

- On simple diagrams, like the two rounds of star graph stabilizer measurements here, we get exactly enough information to do corrections



Detection regions on Steane code

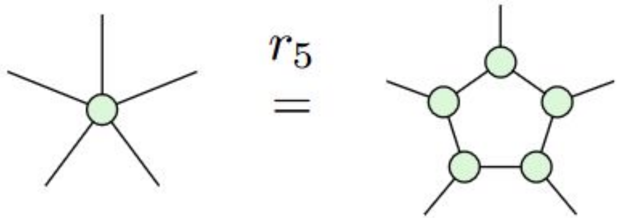
- We obtain a generating set of detection regions
- Kind of fake, because we can’t actually do weight-four measurements
  -> Dist-preserving rewrites

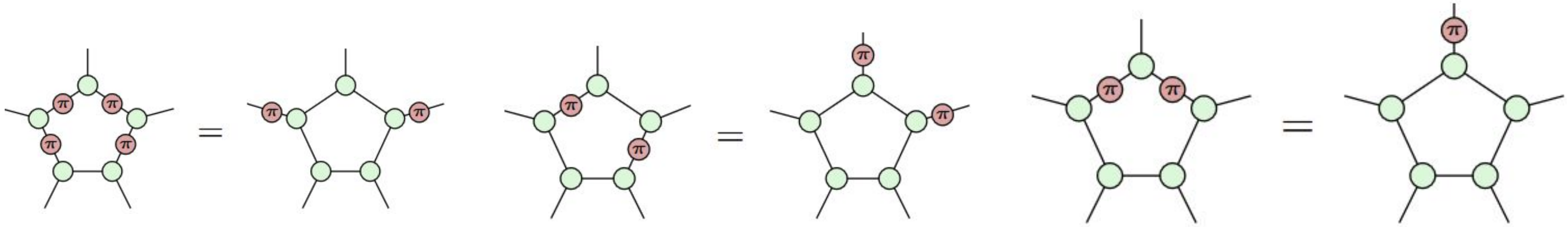


Recap

- A rewrite is distance preserving if the number of non-detectable errors does not increase when pushing errors to the boundary
  -> New diagrams get fault-tolerance

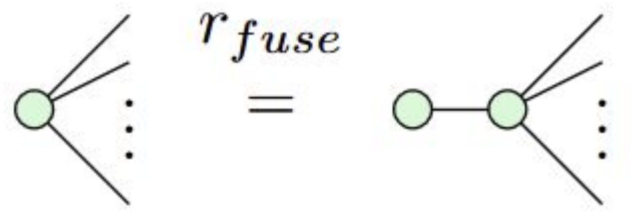
Example dist. preserving rewrite

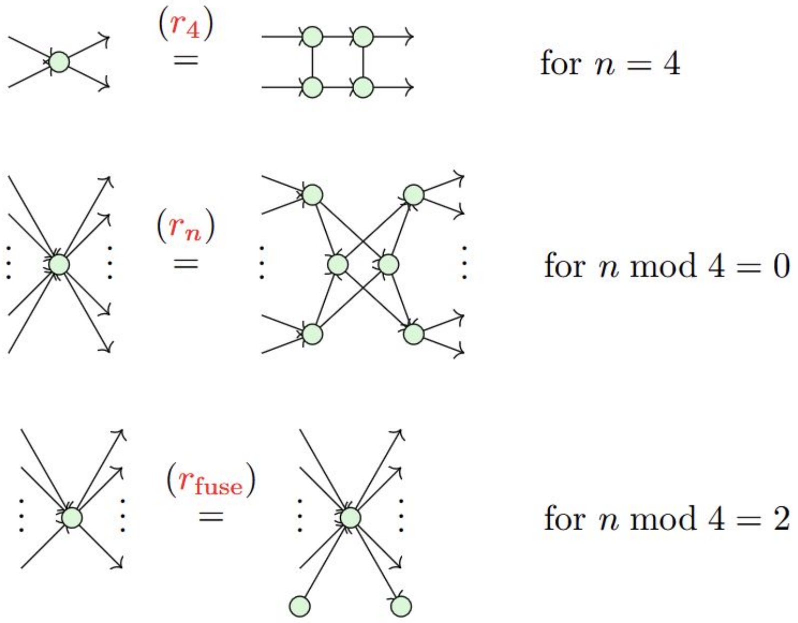




Distance-preserving rewrites

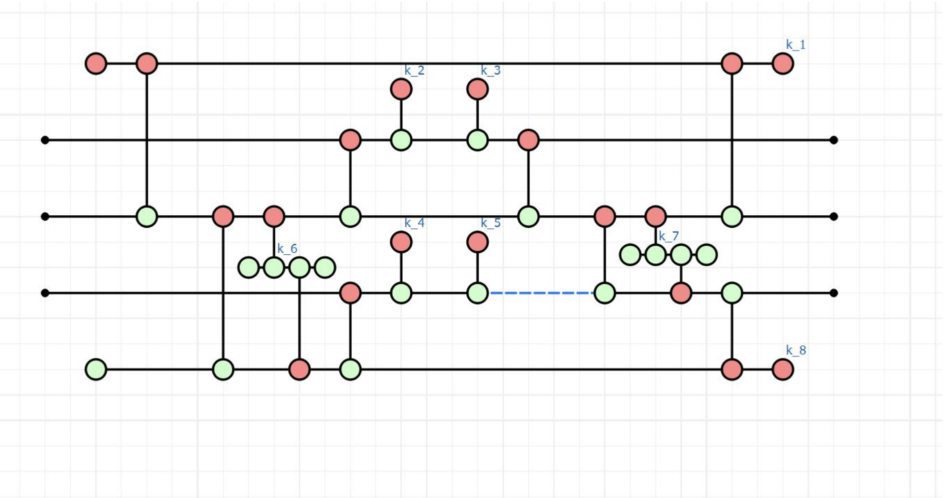
- Distance preserving rewrites lead to no additional spiders upon propagation





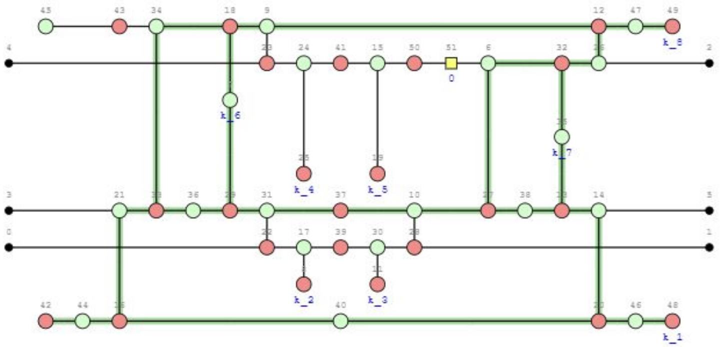
New diagrams

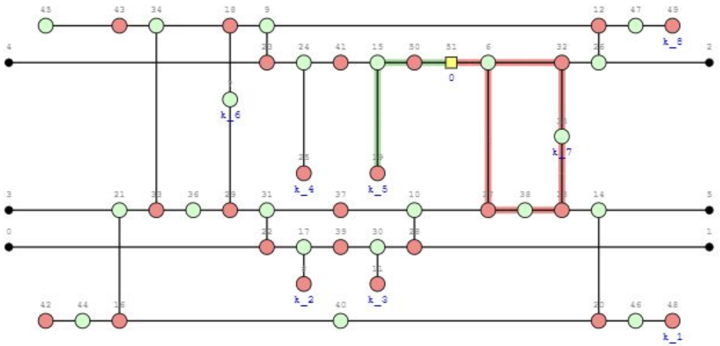
- Using these rewrites, we get new diagrams from which circuit extraction is simple
- But how to interpret the measurement results/what corrections do they imply?



So what do these detection webs on new codes look like?

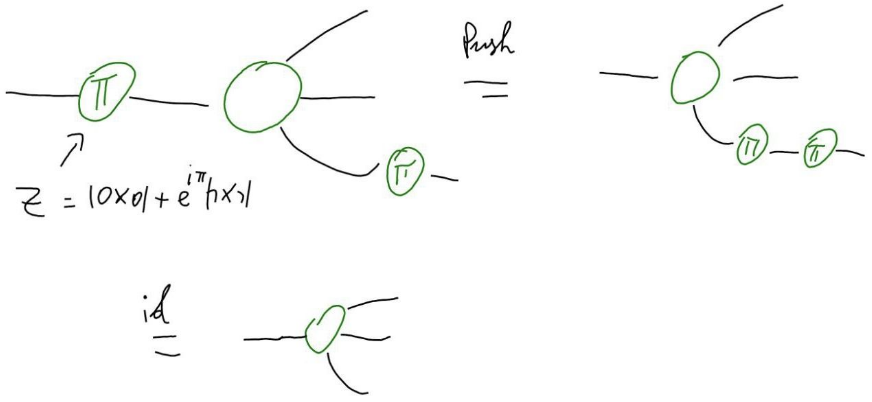
- We find detection webs, however they are not minimal yet



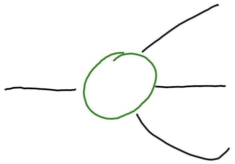


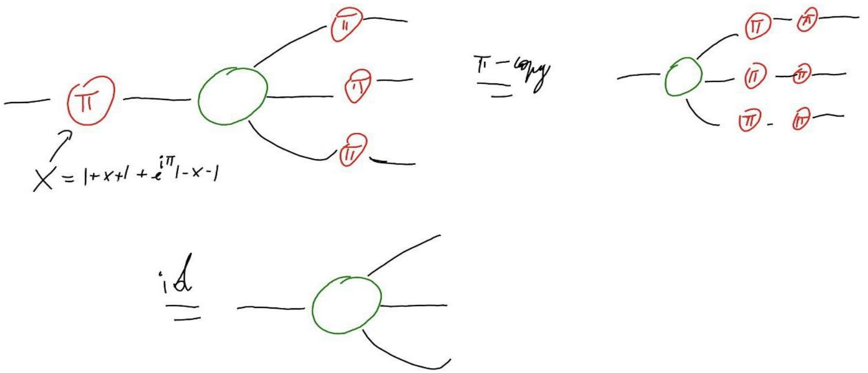
What does it mean to fire a node?

- The diagram of a phaseless node is ZX equivalent to that of node + pi spiders in specific configurations
- This gives two rules:
  - Fire even in same color
  - Fire all or none in opposite color



$$|000\rangle\langle 0| + |111\rangle\langle 1|$$

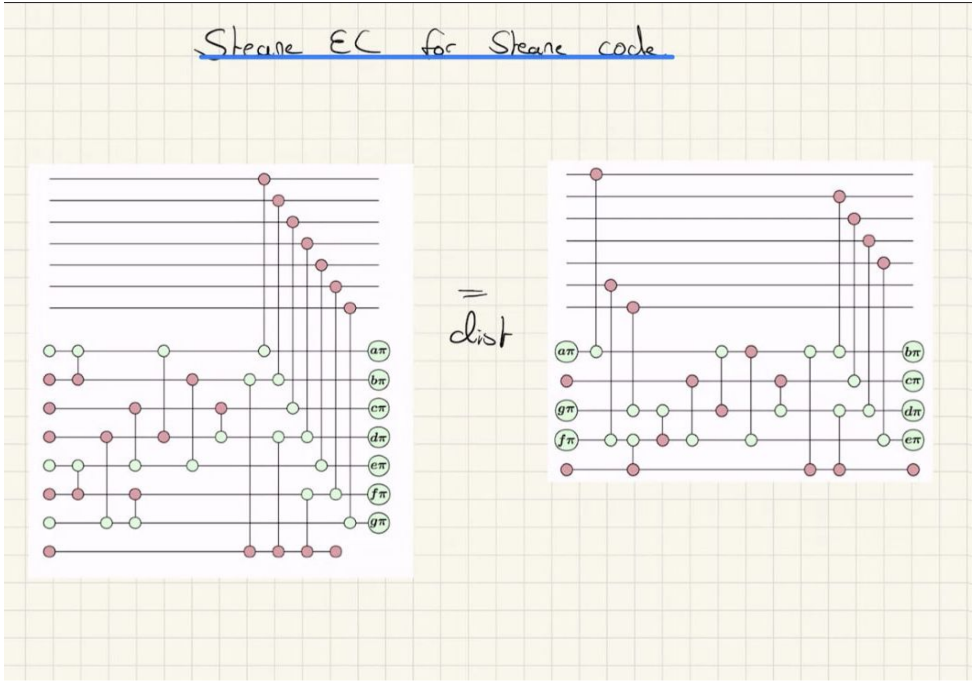




Further cool stuff about fault tolerance and ZX

- Using distance preserving rewrites, borghans recently found a reduced fault-tolerant version of the steane code stabilizer measurements

- Left is previously known fault-tolerant scheme, right is new one using dist-preserving rewrites



Detection regions on this

- The overlap of different detection regions will identify detectable errors and their space-time locations