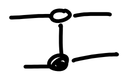
Lecture 7 Clifford diagrams and circuits

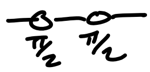
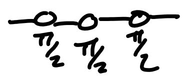
Def A ZX-diagram is <u>Clifford</u> when it is made of <u>Clifford spiders</u>

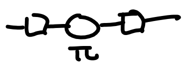
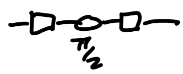
 $k \in \mathbb{Z}$.

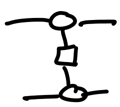
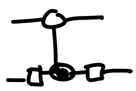
Def <u>Clifford circuits</u> are circuits made from:

$S =$  $H =$  $\approx$  CNOT $=$ 

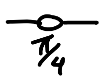
Ex Some common Clifford gates:

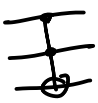
$Z =$  $=$  $S^\dagger =$  $=$ 

$X =$  $=$  $\sqrt{X} =$  $=$ 

$CZ =$  $=$ 

Ex Some <u>non</u>-Clifford gates:

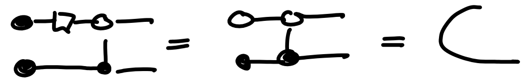
$T =$  $(S = T^2)$

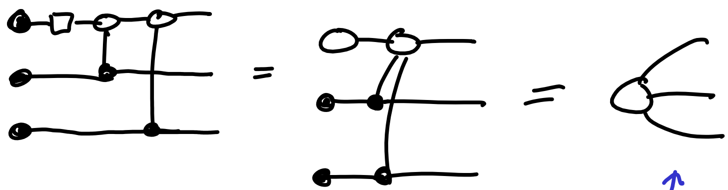
TOF $=$  $= \dots$

Def A *Clifford state* is a state $|\psi\rangle = C|0\dots0\rangle$ for a Clifford circuit C.

Q: Why care about Cliffords?

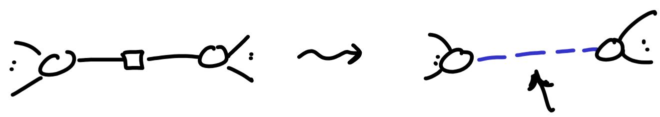
* Contains useful states, e.g.

Bell $\approx$ 

GHZ $\approx$ 

q. nonlocality in PQP

* Quantum error correction (later)
* Eff. classical simulation (soon)
* Rich rewrite theory (now!)

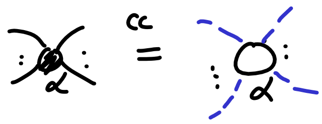
Notation: 

Hadamard edge

Def A ZX diagram is *graph like* if:

1. all spiders are Z spiders
2. all edges btw spiders are Hadamard edges
3. no parallel edges or self-loops
4. every input/output is connected to a spider.

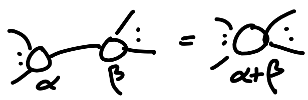
Prop Every ZX-diagram is equal to a graph-like one.

Pf 1. Use 

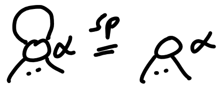
to elim X spiders.

& use 

to cancel extra H's.

2. Use (sp) to elim non-H edges: 

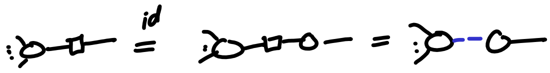
3. For parallel H-edges:

For self-loops: 

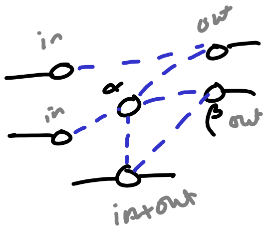
4. Use 

if necessary.

e.g.  ← g.l.

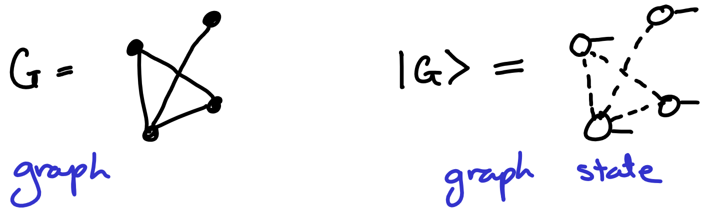


□

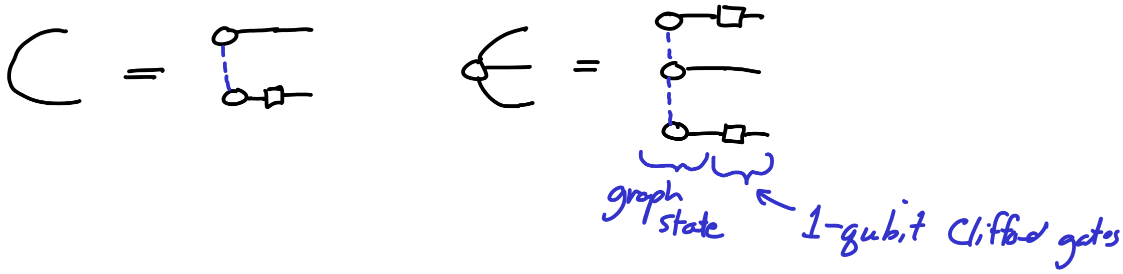
Ex 

Def A graph-like diagram is called a *graph state* if:
- no inputs
- no interior spiders
- no phases

Ex



Some states are almost graph states, e.g.

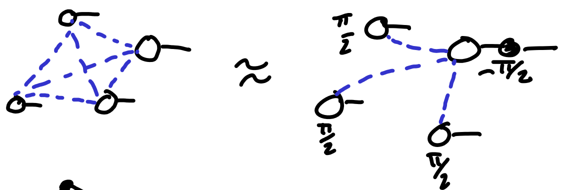


Def A graph state with local Cliffords (GSLC) is a state of the form $(U_1 \otimes \ldots \otimes U_n)|G\rangle$ for some graph state $|G\rangle$ and 1-qubit Clifford gates U_i.

Thm Any Clifford state is = to a GSLC.

We'll need some new tools to prove this!

First, note that for GSLCs, the graph can be deceiving!



Lecture 8

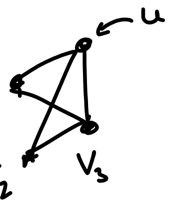
Def Let $G=(V,E)$ be a graph and $u \in V$.

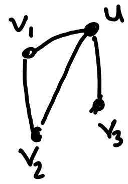
The Local complementation of G about u is a new graph $G*u=(V,E')$ where

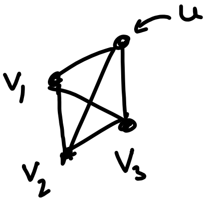
$$\forall v,w \in N_G(u). \quad (v,w)\in E' \Leftrightarrow (v,w)\notin E.$$

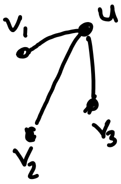
↑ neighbourhood

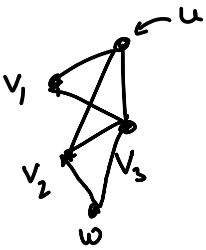
Ex

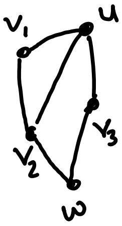
1. $G=$ 

$G*u=$ 

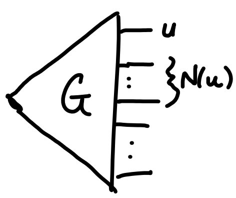
2. $G=$ 

$G*u=$ 

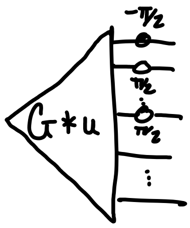
3. $G=$ 

$G*u=$ 

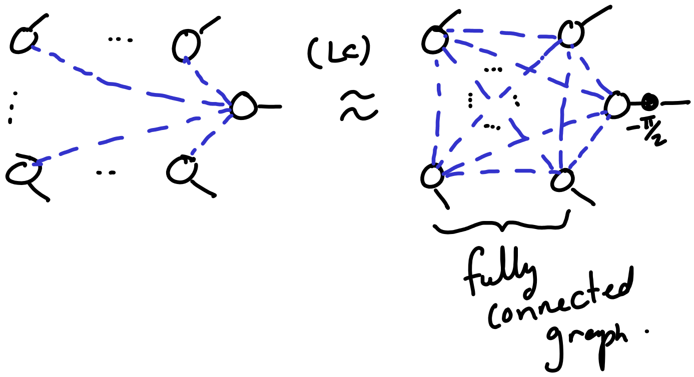
Prop



$\approx$

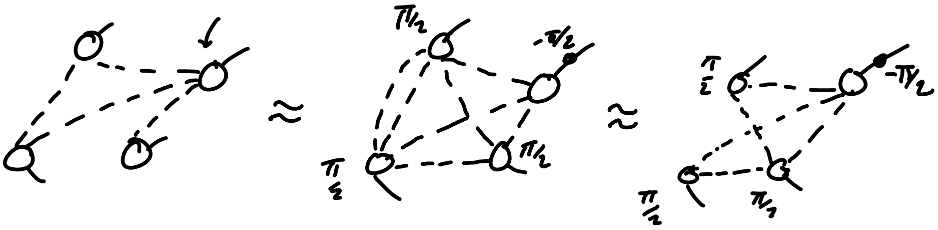


Graphically:

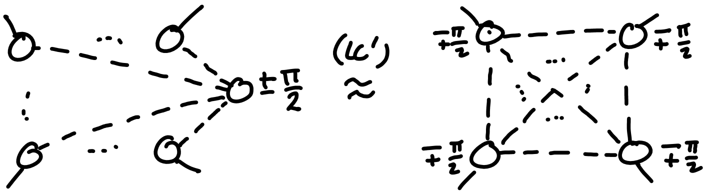


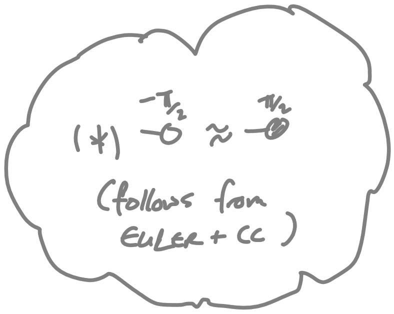
Q: Why is this the same as local comp?

A: Because 

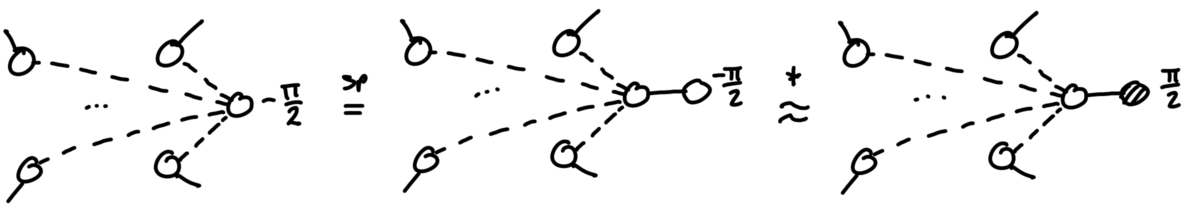


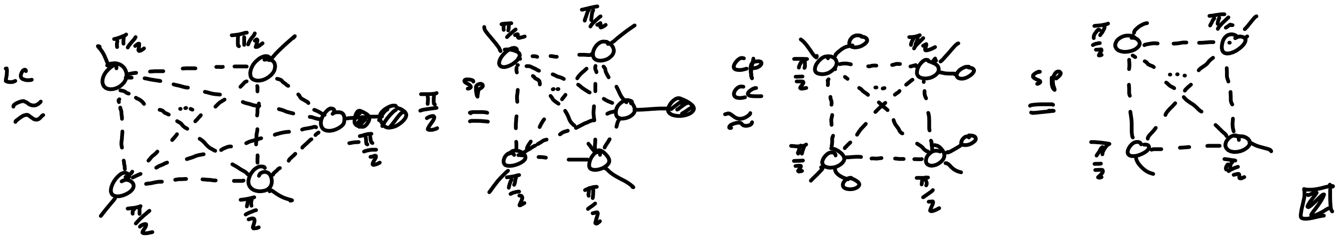
<u>Prop</u>



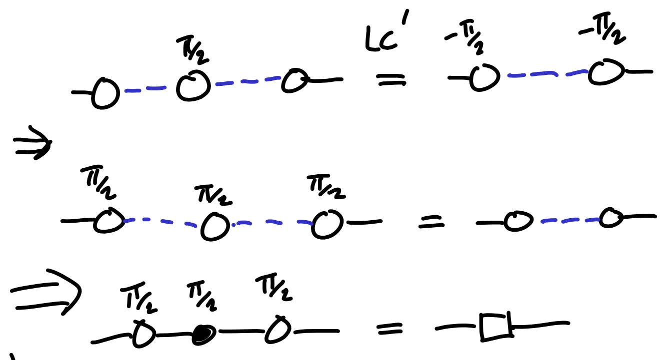


<u>Pf</u>





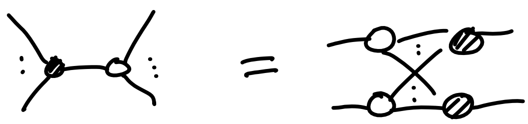
$LC' \Rightarrow EU$



(nb. $EU \Rightarrow LC'$, but harder)

Pivoting.

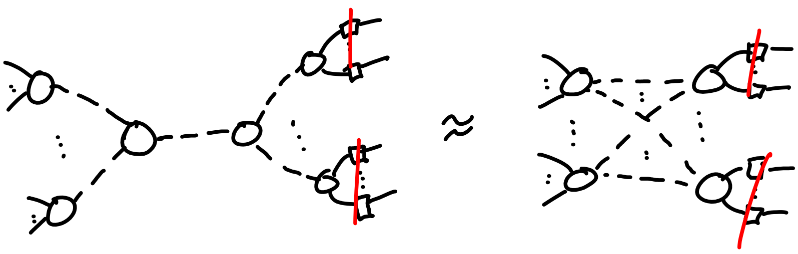
Consider the (sc) rule:

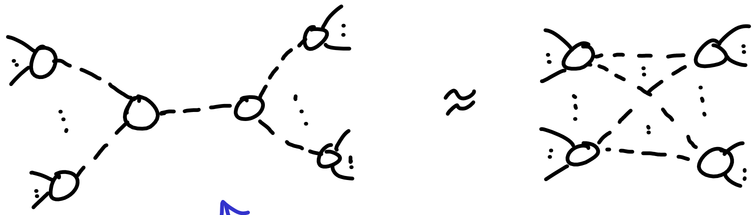


add some context:

always deletes spiders!

Now, (cc) both sides to elim X spiders:

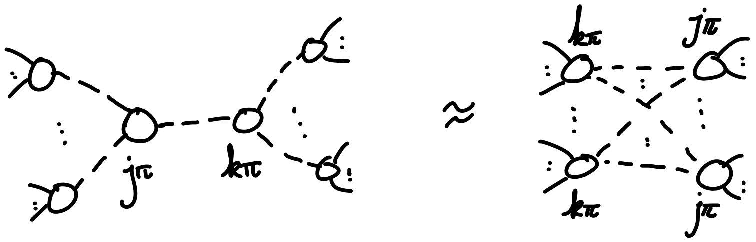




deletes 2 adj. phase-free spiders.

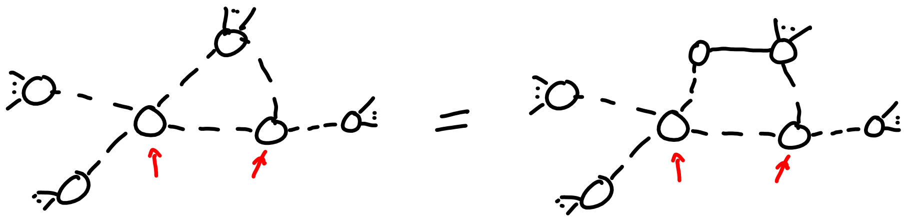
Generalisation:

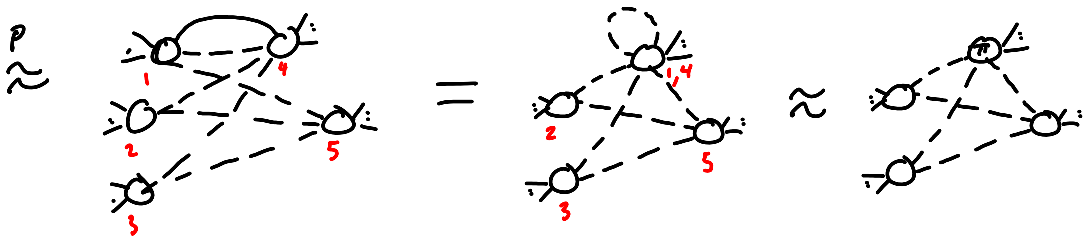
Pivot rule:



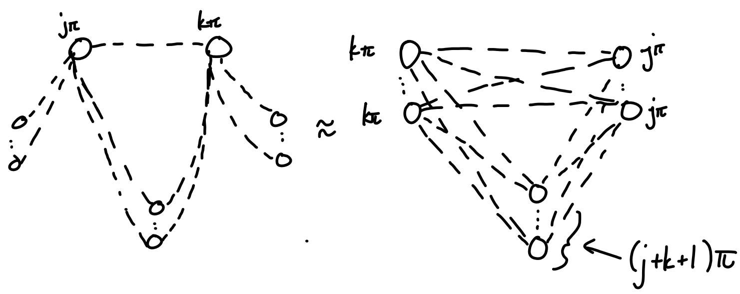
deletes 2 adjacent Pauli spiders.

Q: What if they share neighbours?

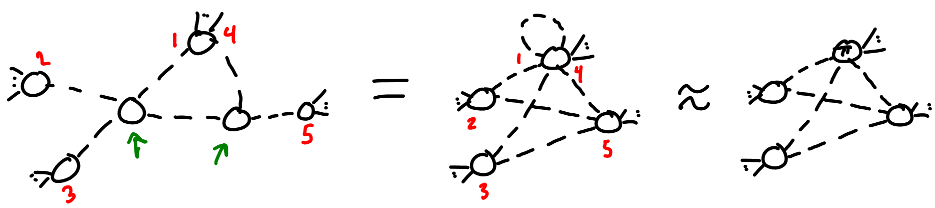




General pivot:



Or, as I prefer to think about it, use the simpler rule, but allow boundary sp's to match twice:



Rewrite strategy (Clifford-simp.)

1. convert to a graph-like diagram
2. apply LC' & P' as long a possible.
3. remove isolated $\{0,\pi\}$-spiders.

PROP1 Clifford-simp terminates for any ZX-diag and removes all interior:

* $\pm\frac{\pi}{2}$ spiders
* pairs of connected $\{0,\pi\}$-spiders

Recall: $m \vdots \boxed{} \vdots n$ is a $2^n \times 2^m$ matrix.

$m = n = 0 \Rightarrow 2^0 \times 2^0 = 1 \times 1$ matrix (a scalar)

Def A scalar ZX-diagram is a ZX-diag w/ no inputs and no outputs.

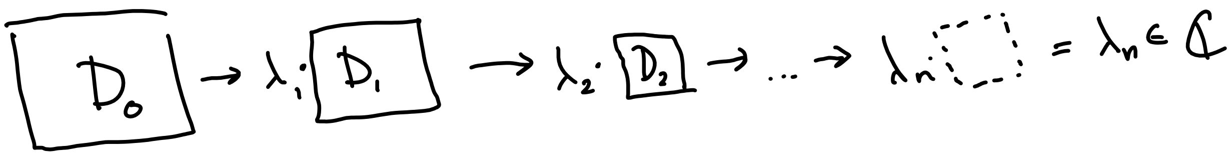
COR (to PROP1) There exists a terminating rewrite strategy that removes all spiders from a scalar Clifford diagram.

Pf First apply Clifford-simp. Then the only spiders left are $\bigcirc$ and $\overset{\pi}{\bigcirc}$. For these:

$$\bigcirc \to 2 \cdot \boxed{} \qquad \overset{\pi}{\bigcirc} \to \emptyset \qquad \blacksquare$$

Q: What's left?

A: the scalar factor



Application 1 (Efficient) strong simulation of Clifford circuits.

Problem For a circuit C, compute:

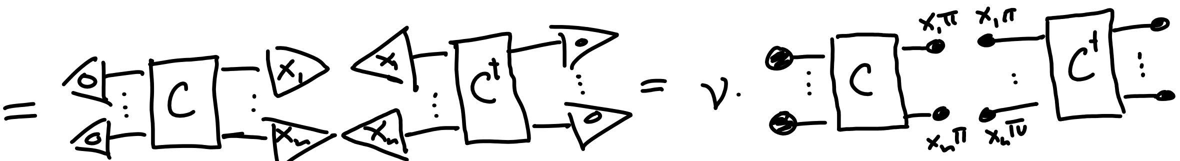
$$(*)\quad \mathrm{Prob}(x_1 \dots x_n \mid |\psi\rangle) \quad \text{where } |\psi\rangle = C|0\dots0\rangle.$$

... or more generally, for $k \leq n$, compute the marginal probability:

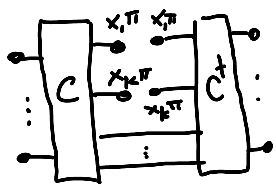
$$(**)\quad \mathrm{Prob}(x_1 \dots x_k \mid |\psi\rangle) = \sum_{x_{k+1},\dots,x_n} \mathrm{Prob}(x_1 \dots x_n \mid |\psi\rangle)$$

Born rule

$$\mathrm{Prob}(x_1 \dots x_n \mid |\psi\rangle) := \left| \langle x_1 \dots x_n | C | 0 \dots 0\rangle \right|^2$$



$$\Rightarrow \mathrm{Prob}(x_1 \dots x_k \mid |\psi\rangle) = \sum_{x_{k+1} \dots x_n} \mathrm{Prob}(x_1 \dots x_n \mid |\psi\rangle) = \nu' \cdot$$



ZX-diagram of (**)

$\left(\sum_x \right.$  $\left. = 2 \cdot \sum_x |x\rangle\langle x| = 2I \right)$

Lecture 9

Algorithm 1: For a circuit C:

1. Let D be the ZX-diagram of $\mathrm{Prob}(x_1 .. x_k \mid C|0...0\rangle)$.
2. Apply Clifford-simp to get a number.

Prop 1 Algorithm 1 terminates in polynomial time (in the # of qubits & gates of C).

Pf Assume basic diagram operations (add/remove spider/wire) take constant time. If C has n qubits & k gates, D has at most $S := 2\cdot(2n+2k) = 4(n+k)$ spiders. Then:

— Each rewrite removes 1 or 2 spiders, so there are at most $4(n+k)$ steps.

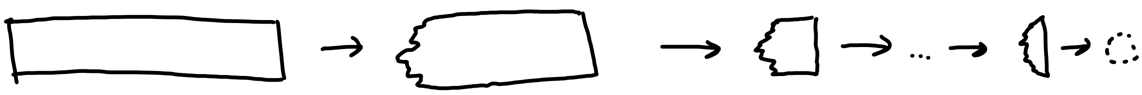
— Each step adds/removes at most $(4(n+k))^2$ edges, so Algorithm 1 performs $O((n+k)^3)$ basic graph operations. ∎

Rem this is not optimal. A good choice of LC' and P' steps actually takes $O(n^2 k)$ time.
$\Rightarrow$ if $k \gg n$, this makes a big difference!

IDEA:

1. Avoid big spiders:

2. Apply LC' & P' from left-to-right:



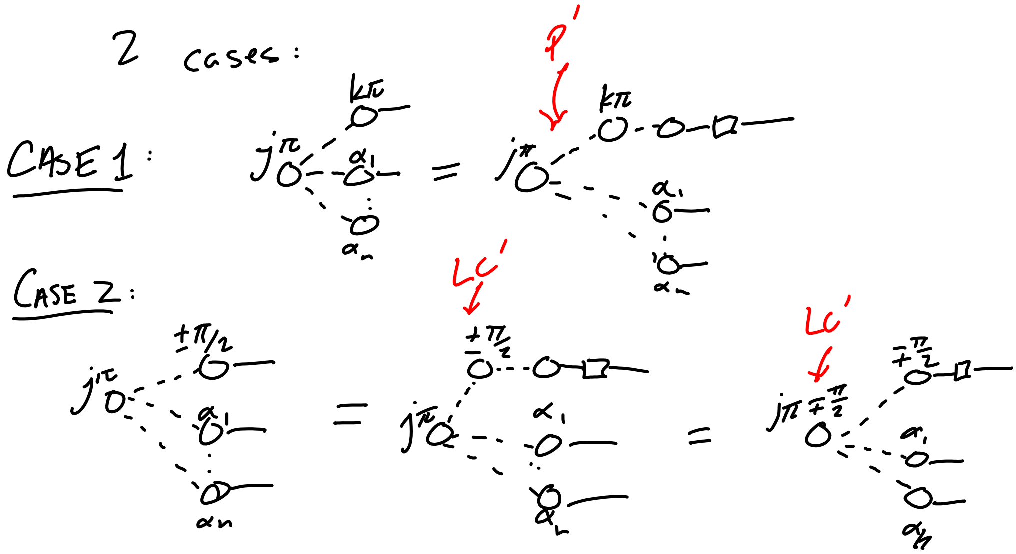
$\Rightarrow$ each step involves at most $O(n)$ spiders (hence $O(n^2)$ wires)

Def A graph-like ZX-diagram is in AP-form if all interior spiders:
- have phase $\in 0, \pi$
- are only connected to boundary spiders.

Def A ZX-diagram is in graph-state w/ local Clifford (GSLC) form if it has
* all Z spiders, fused as much as possible
* all spiders are connected to exactly 1 input (possibly via a 1-qubit Clifford unitary)

AP $\rightarrow$ GSLC :

2 cases:



<u>Application 2</u> Efficient synthesis of Clifford circuits.

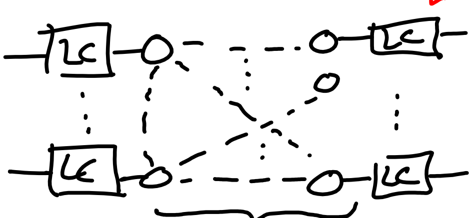
Clifford diagram $\rightarrow$ AP-form $\longrightarrow$ GSLC

* only internal spiders are $k\pi$ * <u>no</u> internal spiders.

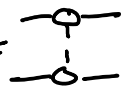
Algorithm 2 (Clifford re-synthesis)

1. For an n-qubit Clifford circuit C, translate to ZX-diagram D.
2. Compute GSLC form.
3. Write as:

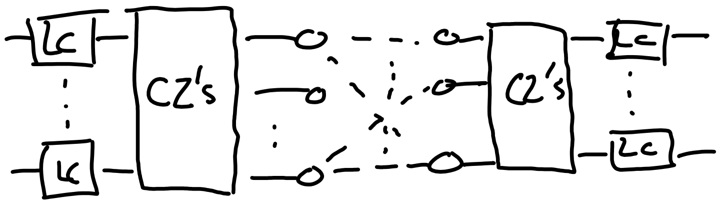
$O(n)$



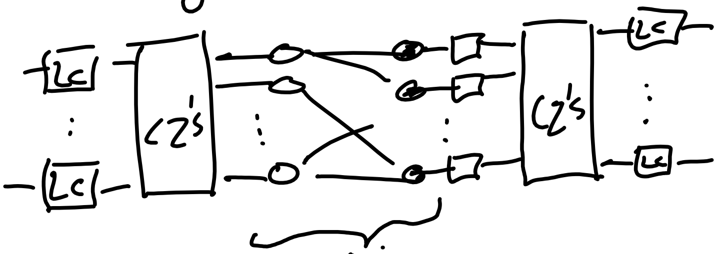
graph "state"

4. unfuse CZ gates =  and

$O(n^2)$

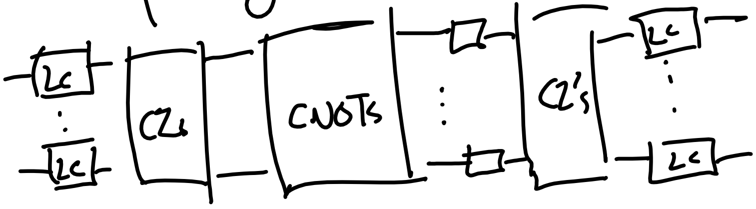


5. colour-change:



4. extract parity map as CNOTs

$O(n^2)$



Prop Any Clifford circuit can be written w/ at most $O(n^2)$ gates!