Counting perfect matchings with the planar fragment of ZW-Calculus

Titouan Carette, Etienne Moutot,
Thomas Perez, and Renaud Vilmart

Motivation

- Recent work : Compositionality of planar perfect matchings.

Motivation

- Recent work : Compositionality of planar perfect matchings.
- We introduced the planar fragment of ZW-calculus (**pW**-calculus) and showed it is **universal** and **complete** for **matchgates**.

Motivation

- Recent work : Compositionality of planar perfect matchings.
- We introduced the planar fragment of ZW-calculus (**pW**-calculus) and showed it is **universal** and **complete** for **matchgates**.
- **pW**-scalars correspond to the number of perfect matchings of graphs.

Motivation

- Recent work : Compositionality of planar perfect matchings.
- We introduced the planar fragment of ZW-calculus (**pW**-calculus) and showed it is **universal** and **complete** for **matchgates**.
- **pW**-scalars correspond to the number of perfect matchings of graphs.
- **FKT algorithm** : efficiently simulable fragment (like Clifford fragment for the ZX-calculus)

Motivation

- Recent work : Compositionality of planar perfect matchings.
- We introduced the planar fragment of ZW-calculus (**pW**-calculus) and showed it is **universal** and **complete** for **matchgates**.
- **pW**-scalars correspond to the number of perfect matchings of graphs.
- **FKT algorithm** : efficiently simulable fragment (like Clifford fragment for the ZX-calculus)
- **In this talk** : alternative diagrammatical polynomial algorithm counting perfect matchings in planar graphs using **pW**-calculus

1. Perfect Matchings
2. ZW-calculus, **pW**-calculus
3. Rewriting strategy
4. A diagrammatical algorithm
5. Further work

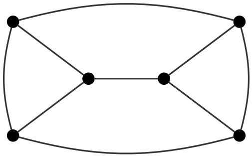
Summary

Perfect matchings

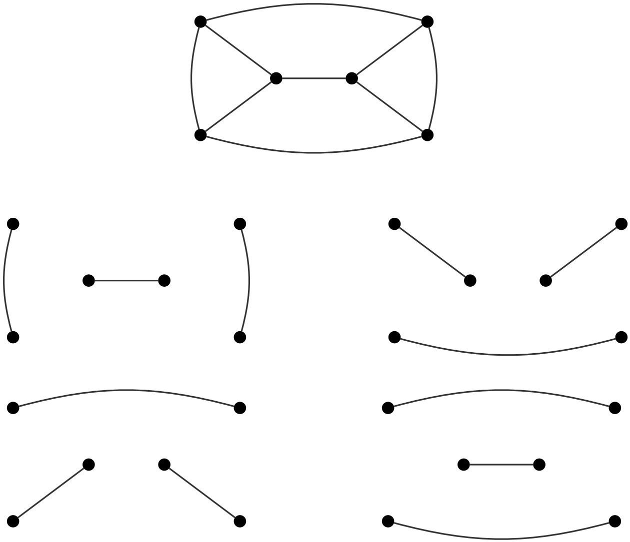
Definition

Let $G = (V, E)$ be a graph.
We say that $P \subset E$ is a perfect matching when every vertex is covered by exactly one edge in P.

Example



Example



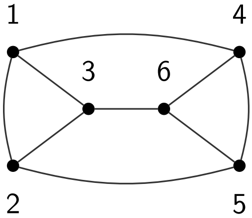
Counting perfect matching

- In general, $\#P$-Complete [Valiant, '79].

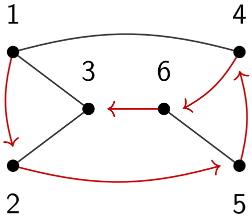
Counting perfect matching

- In general, $\#P$-Complete [Valiant, '79].
- Polynomial for planar graphs [Fisher-Kasteleyn-Temperley, '67]

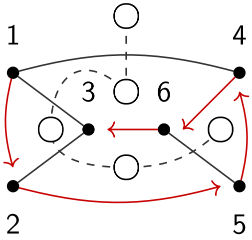
FKT algorithm example



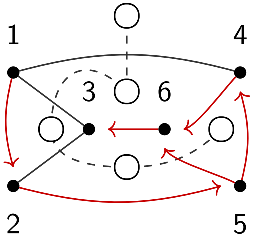
FKT algorithm example



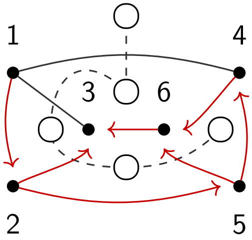
FKT algorithm example



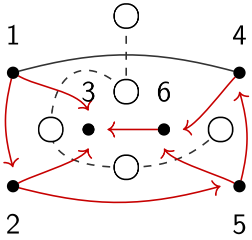
FKT algorithm example



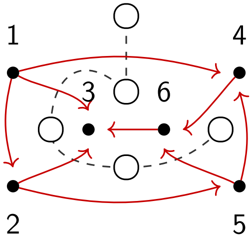
FKT algorithm example



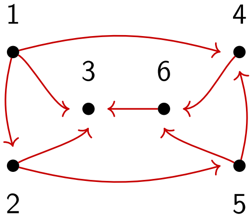
FKT algorithm example



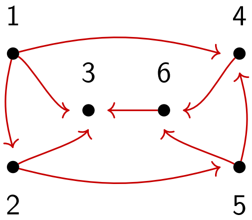
FKT algorithm example



FKT algorithm example



FKT algorithm example



$$\#PM = Pf(M) = \sqrt{det(M)} = 4$$

Summary

ZW-calculus

- Bob Coecke and Aleks Kissinger : The compositional structure of multipartite quantum entanglement.

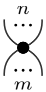
ZW-calculus

- Bob Coecke and Aleks Kissinger : The compositional structure of multipartite quantum entanglement.
- Amar Hadzihasanovic : A diagrammatic axiomatisation for qubit entanglement.

ZW-calculus

- Bob Coecke and Aleks Kissinger : The compositional structure of multipartite quantum entanglement.
- Amar Hadzihasanovic : A diagrammatic axiomatisation for qubit entanglement.
- Interesting connections with fermionic quantum computing, and recent works importing some ZW-calculus primitives into ZX-calculus

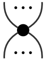
Planar W-calculus



$$:= \sum_{\substack{u\in\{0,1\}^m\\ v\in\{0,1\}^n\\ |uv|=1}} |u\rangle\langle v|$$

$$:= |0\rangle\langle 0| + r\,|1\rangle\langle 1|$$

Planar W-calculus



$$:= \sum_{\substack{u\in\{0,1\}^m \\ v\in\{0,1\}^n \\ |uv|=1}} |u\rangle\langle v|$$

$$:= |0\rangle\langle 0| + r\,|1\rangle\langle 1|$$

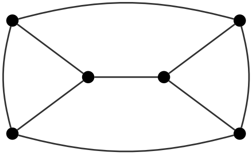
$$:= \langle 00| + \langle 11|$$

$$:= |00\rangle + |11\rangle$$

Correspondence

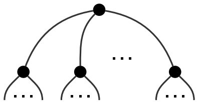
The interpretation of the **pW**-scalar associated to a graph is its number of perfect matchings.

Example

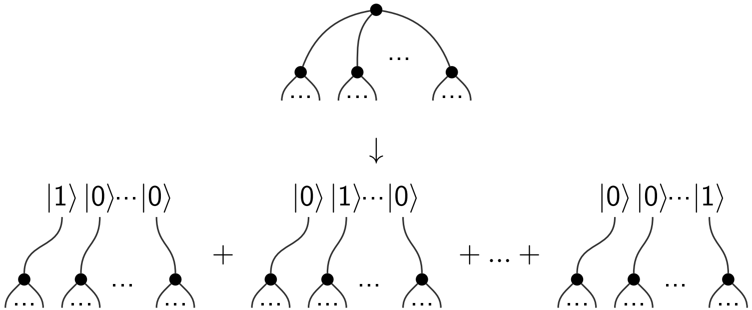


$= 4$

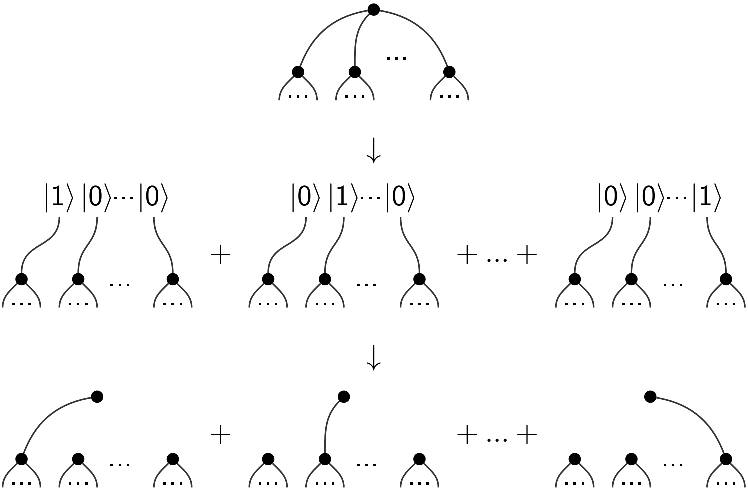
Vertex by vertex Decomposition



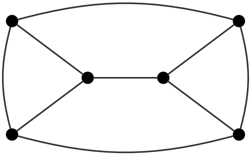
Vertex by vertex Decomposition



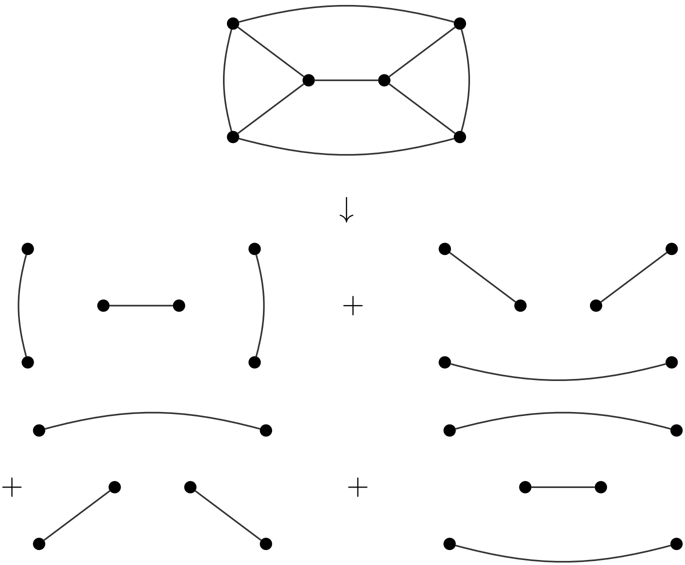
Vertex by vertex Decomposition



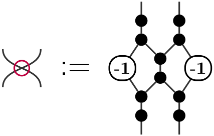
Example



Example



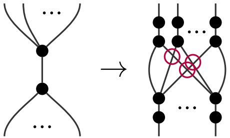
Fermionic Swap

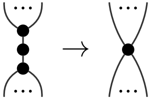


$$:= \sum_{x,y\in\{0,1\}} (-1)^{xy} \, |xy\rangle\langle yx|$$

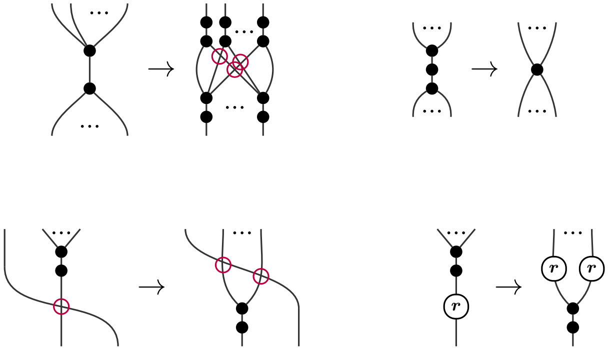
Summary

Rewriting Strategy

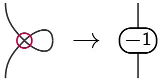


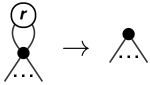


Rewriting Strategy

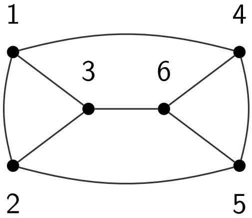


Rewriting Strategy

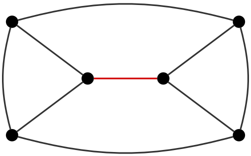




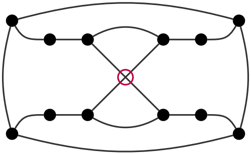
Example



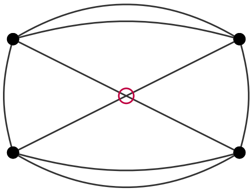
Example



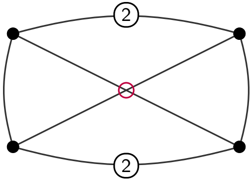
Example



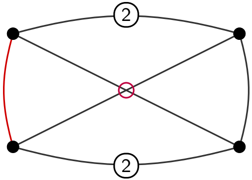
Example



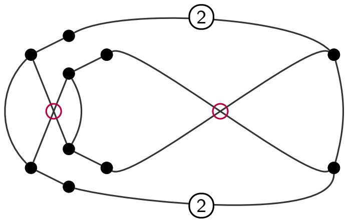
Example



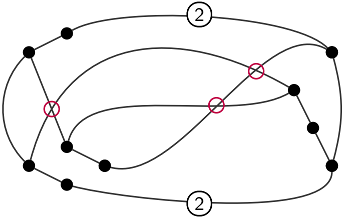
Example



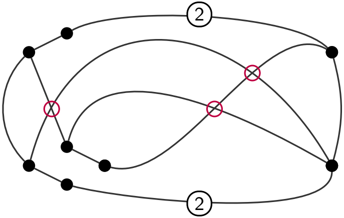
Example



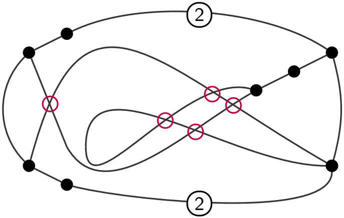
Example



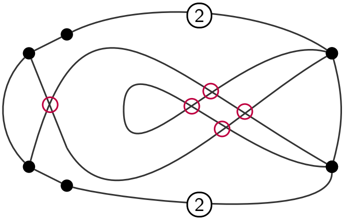
Example



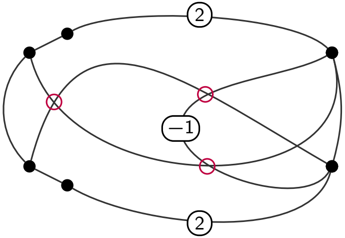
Example



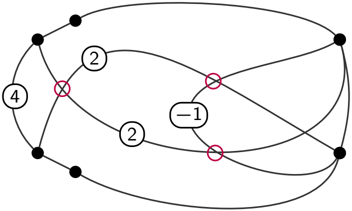
Example



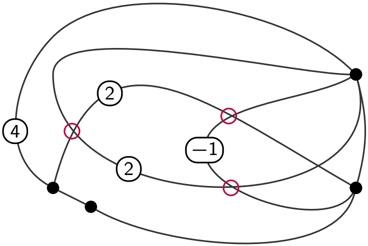
Example



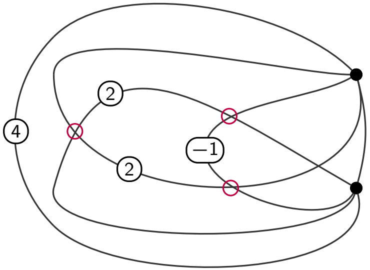
Example



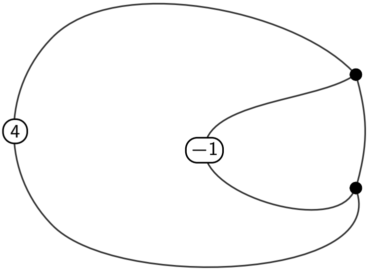
Example



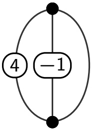
Example



Example



Example



Example

Summary

Embedding

- **Problem** : unclear where one should apply the rewriting rules, why it would terminate, how to implement it.

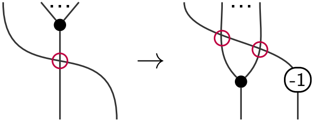
Embedding

- **Problem** : unclear where one should apply the rewriting rules, why it would terminate, how to implement it.
- **Solution** : Let's embed the diagram associated to our graph on a semi-circle.

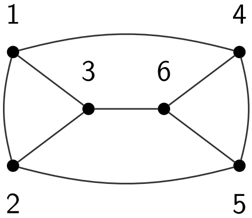
Embedding

- **Problem** : unclear where one should apply the rewriting rules, why it would terminate, how to implement it.
- **Solution** : Let's embed the diagram associated to our graph on a semi-circle.
- **Advantage** : One can consider the nodes ordered from left to right, use this order to know on which edge to apply the rewriting strategy, and deduce whether or not two wires intersect each other via a fermionic swap from the order on their vertices.

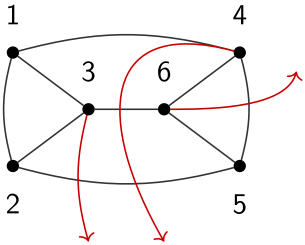
Embedding a diagram



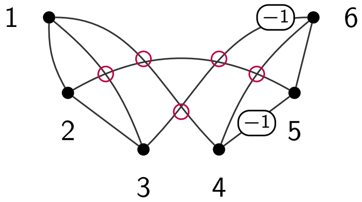
Example



Example



Example



- **Question** : How to embed our diagram in practice ?

- **Question** : How to embed our diagram in practice ?
- **Solution** : We don't need to do it graphically, and can use a Pfaffian orientation to put -1 weights on the right edges.

Proposition

Proposition

The interpretation of a diagram associated to a graph embedded on a semi-circle is the Pfaffian of its adjacency matrix.

Pfaffian

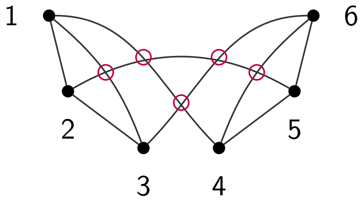
Definition

Let $M \in \mathbb{C}^{2n \times 2n}$,

$$Pf(M) := \sum_{\substack{\pi \in S_{2n} s.t. \forall i \\ \pi(2i-1) < \pi(2i) \\ \pi(2i-1) < \pi(2i+1)}} \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$

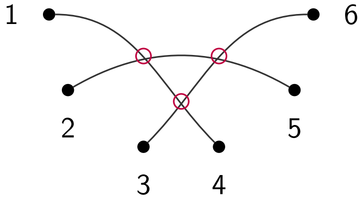
Sketch of proof

$$Pf(M) := \sum_{\substack{\pi \in S_{2n} s.t. \forall i \\ \pi(2i-1)<\pi(2i) \\ \pi(2i-1)<\pi(2i+1)}} \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



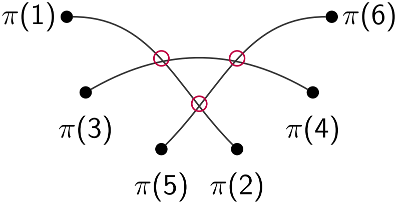
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



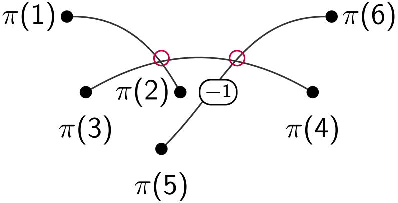
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



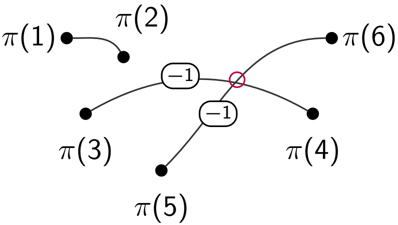
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix}: \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



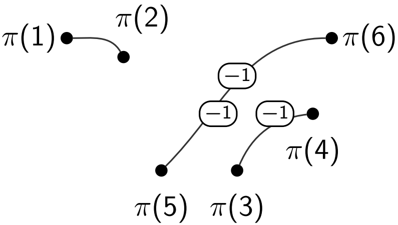
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



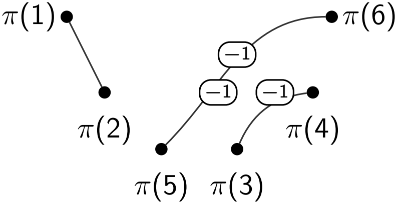
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



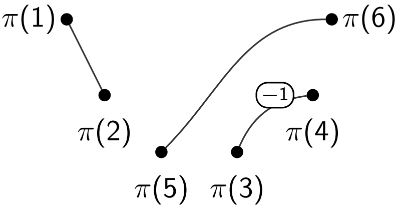
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



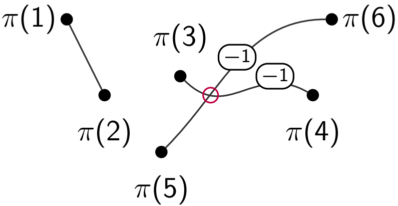
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



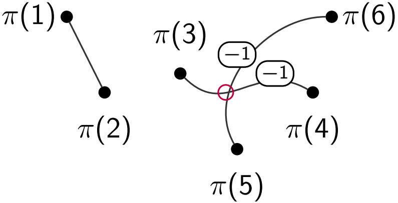
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



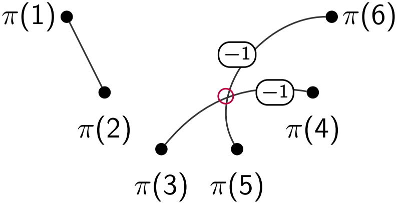
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



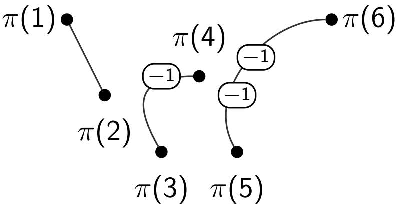
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



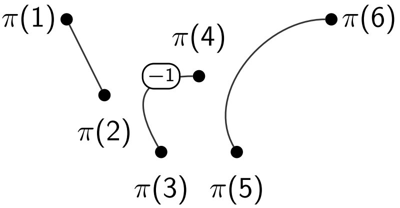
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



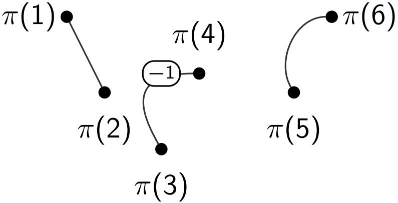
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



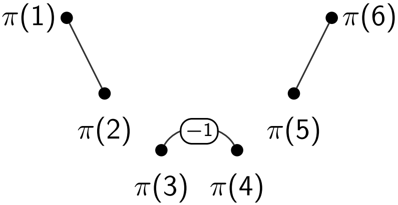
Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix}: \qquad \epsilon(\pi)\prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



Sketch of proof

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix} : \qquad \epsilon(\pi) \prod_{i=1}^{n} M_{\pi(2i-1),\pi(2i)}$$



Algorithm

1. Find a Pfaffian orientation

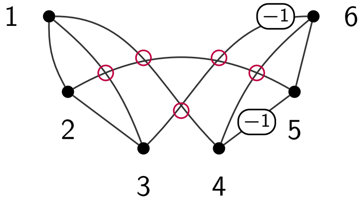
Algorithm

1. Find a Pfaffian orientation
2. Embed the diagram using this orientation

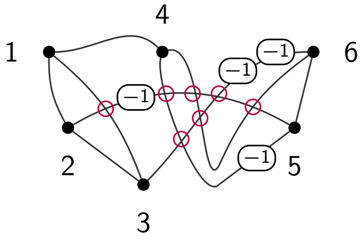
Algorithm

1. Find a Pfaffian orientation
2. Embed the diagram using this orientation
3. While there is more than 2 black spiders, apply the rewriting strategy on edge (i, j), i being minimal and j maximal

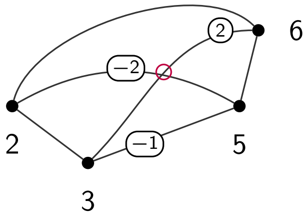
Example



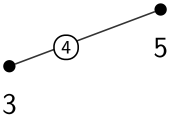
Example



Example



Example



A new algorithm ?

- We have a new graphical algorithm for computing the Pfaffian,

A new algorithm ?

- We have a new graphical algorithm for computing the Pfaffian,
- But still use FKT algorithm to get a Pfaffian orientation and embed our diagram.

A new algorithm ?

- We have a new graphical algorithm for computing the Pfaffian,
- But still use FKT algorithm to get a Pfaffian orientation and embed our diagram.
- Gives a graphical interpretation of the role of the Pfaffian oriantation.

Summary

Further work

- Adapt the algorithm to compute scalars in the fermionic ZW-calculus (**pW** $+$ swap gate) which is universal for **Qubits** modulo an encoding trick.

Further work

- Adapt the algorithm to compute scalars in the fermionic ZW-calculus (**pW** + swap gate) which is universal for **Qubits** modulo an encoding trick.
- Extend the polynomial algorithm to all graphs for which counting perfect matchings is polynomial.

Further work

- Adapt the algorithm to compute scalars in the fermionic ZW-calculus (**pW** + swap gate) which is universal for **Qubits** modulo an encoding trick.
- Extend the polynomial algorithm to all graphs for which counting perfect matchings is polynomial.
- Explore the link between Pfaffian orientations and planar embeddings of a graph.

The end

Thank you for your attention