

The Proof Complexity of Matrix Algebra

Michael Soltys

McMaster University, Canada

1. Quantified Permutation Frege (with Tim Paterson)
2. Steinitz Exchange Lemma
3. Clow Sequences

$$(\exists ab)\alpha \quad \equiv (\alpha(a, b) \vee \alpha(b, a)) \quad \equiv (\alpha \vee \alpha^{(ab)})$$

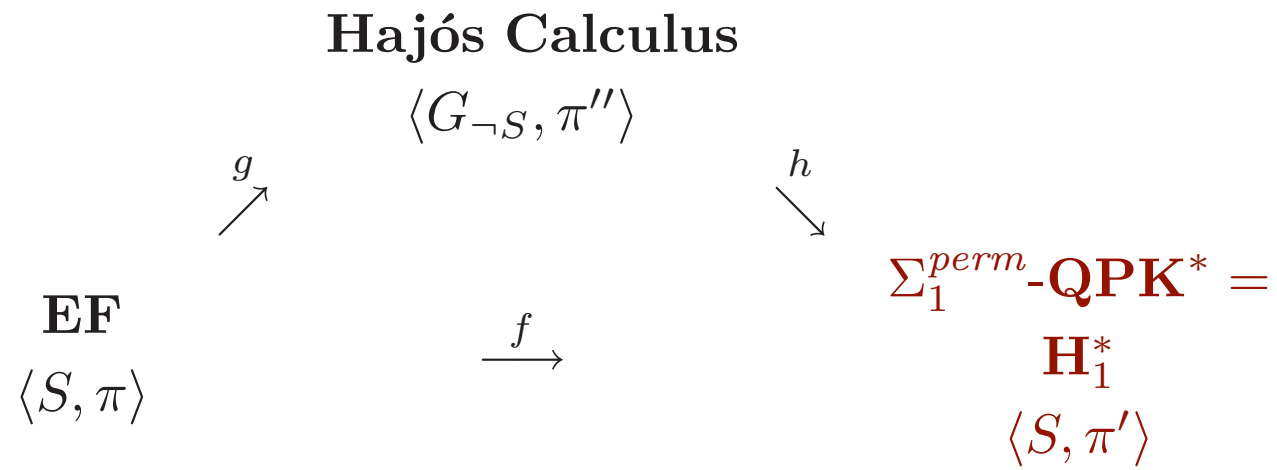
$$(\forall ab)\alpha \quad \equiv (\alpha(a, b) \wedge \alpha(b, a)) \quad \equiv (\alpha \wedge \alpha^{(ab)})$$

QPK

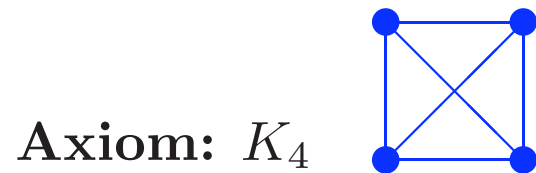
$$\begin{array}{cc}
 \frac{\alpha, \Gamma \rightarrow \Delta}{(\forall ab)\alpha', \Gamma \rightarrow \Delta} & \frac{\Gamma \rightarrow \Delta, \alpha}{\Gamma \rightarrow \Delta, (\exists ab)\alpha'} \\
 \\
 \frac{\alpha, \Gamma \rightarrow \Delta}{(\exists ab)\alpha', \Gamma \rightarrow \Delta} & \frac{\Gamma \rightarrow \Delta, \alpha}{\Gamma \rightarrow \Delta, (\forall ab)\alpha'}
 \end{array}$$

α' is α or $\alpha^{(ab)}$

Restriction: for every $\beta \in \Gamma \cup \Delta$, $(ab) \in \text{Aut}(\beta)$, i.e., $\beta \equiv \beta^{(ab)}$.

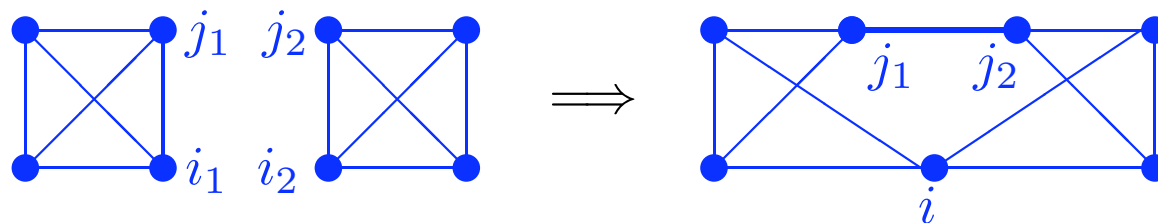


Hajós Calculus



Addition Rule: Add any number of vertices and/or edges.

Join Rule:



Contraction Rule: Contract two nonadjacent vertices into a single vertex, and remove the resulting duplicated edges.

We can express k -colorability of graphs in $\exists\mathbf{PLA}$.

Let 0_i denote the $i \times i$ matrix of zeros.

Let G be a graph, and A_G its adjacency matrix. We can state that G is k -colorable, for any fixed k , as follows:

$$\underbrace{(\exists P)(\exists i_1, i_2, \dots, i_k)}_{\text{bounded by size of } A_G} [PA_G P^t = \left[\begin{array}{c|c|c|c} 0_{i_1} & * & \dots & * \\ \hline * & 0_{i_2} & \dots & * \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline * & * & * & 0_{i_k} \end{array} \right]$$

Let $\text{non3col}(X)$ be the negation of this formula with $k = 3$.

Let $\text{HC}(X, Y)$ be the (\mathbf{LA}) formula stating that Y is a \mathbf{HC} derivation of X .

Theorem: $\forall\mathbf{PLA} \vdash \text{HC}(X, Y) \rightarrow \text{non3col}(X)$.

- $\forall \mathbf{PLA} \vdash \mathbf{HC}(X, Y) \rightarrow \mathbf{non3col}(X)$
- $\mathbf{H}_1^* \vdash_{p(|\sigma|)} \|\mathbf{HC}(X, Y) \rightarrow \mathbf{non3col}(X)\|_\sigma$ and so
 $\mathbf{H}_1^* \vdash_{p(|\hat{\sigma}|)} \|\mathbf{HC}(\langle G_{\neg S} \rangle, \langle \pi'' \rangle)\|_{\hat{\sigma}} \rightarrow \|\mathbf{non3col}(\langle G_{\neg S} \rangle)\|_{\hat{\sigma}}$.
- $\mathbf{H}_1^* \vdash_{p(|\hat{\sigma}|)} \|\mathbf{non3col}(\langle G_{\neg S} \rangle)\|_{\hat{\sigma}} \rightarrow S$

We encode a set of vectors $\{v_1, v_2, \dots, v_n\}$ as a matrix

$$T = [v_1 v_2 \dots v_n].$$

Steinitz Exchange Theorem (SET): if T is *total*, and E is *linearly independent*, then there exists $F \subseteq T$, such that $|F| = |E|$, and $(T - F) \cup E$ is total.

$$\exists X, TX = I \wedge (\forall Y \neq 0, EY \neq 0) \rightarrow \exists F \subseteq T, |F| = |E| \wedge \exists X, (T - F \cup E)X = I$$

So **SET** is a Π_2^B formulas of **QLA**.

Given T, E , the matrix F can be computed in \mathbf{NC}^2 .

Suppose $E = [e_1 e_2]$ and $T = [t_1 t_2 t_3 t_4]$, then consider

$$[e_1 e_2] \quad [e_1 e_2 t_1] \quad [e_1 e_2 t_1 t_2] \quad [e_1 e_2 t_1 t_2 t_3] \quad [e_1 e_2 t_1 t_2 t_3 t_4]$$

Independently for every $i = 0, 1, 2, 3$, if

$$\text{rank}([e_1 e_2 t_1 \dots t_i]) = \text{rank}([e_1 e_2 t_1 \dots t_{i+1}])$$

then put t_{i+1} in F .

Rank can be computed in \mathbf{NC}^2 with *Mulmuley's algorithm*.

Mulmuley's Algorithm^a

M is $n \times n$ matrix, $p_M(x)$ its char. poly. (Berkowitz's alg.)

Geometric Rank : usual rank.

Algebraic Rank : $n -$ (highest power of x that divides $p_M(x)$).

^a*Parallel Linear Algebra*, by Joachim von zur Gathen, chapter in *Synthesis of Parallel Algorithms*.

Mulmuley's Algorithm^a

M is $n \times n$ matrix, $p_M(x)$ its char. poly. (Berkowitz's alg.)

Geometric Rank : usual rank.

Algebraic Rank : $n - (\text{highest power of } x \text{ that divides } p_M(x))$.

Claim: $\text{Rank}_G(M) = \text{Rank}_G(M^2) \Rightarrow \text{Rank}_G(M) = \text{Rank}_A(M)$.

^a*Parallel Linear Algebra*, by Joachim von zur Gathen, chapter in *Synthesis of Parallel Algorithms*.

Mulmuley's Algorithm^a

M is $n \times n$ matrix, $p_M(x)$ its char. poly. (Berkowitz's alg.)

Geometric Rank : usual rank.

Algebraic Rank : $n - (\text{highest power of } x \text{ that divides } p_M(x))$.

Claim: $\text{Rank}_G(M) = \text{Rank}_G(M^2) \Rightarrow \text{Rank}_G(M) = \text{Rank}_A(M)$.

Given M , let M' be

$$\begin{pmatrix} 0 & M^t \\ M & 0 \end{pmatrix}$$

Clearly, $\text{rank}_G(M) = \frac{1}{2}\text{rank}_G(M')$.

^a*Parallel Linear Algebra*, by Joachim von zur Gathen, chapter in *Synthesis of Parallel Algorithms*.

Mulmuley's Algorithm^a

M is $n \times n$ matrix, $p_M(x)$ its char. poly. (Berkowitz's alg.)

Geometric Rank : usual rank.

Algebraic Rank : $n - (\text{highest power of } x \text{ that divides } p_M(x))$.

Claim: $\text{Rank}_G(M) = \text{Rank}_G(M^2) \Rightarrow \text{Rank}_G(M) = \text{Rank}_A(M)$.

Given M , let M' be

$$\begin{pmatrix} 0 & M^t \\ M & 0 \end{pmatrix}$$

Clearly, $\text{rank}_G(M) = \frac{1}{2}\text{rank}_G(M')$.

$M'' = M' \cdot \text{diag}(1, y, y^2, \dots, y^{2n-1})$; $\text{rank}_G(M'') = \text{rank}_G((M'')^2)$.

y is an indeterminate, and M'' is a matrix over the field $\mathbb{F}(y)$.

^a*Parallel Linear Algebra*, by Joachim von zur Gathen, chapter in *Synthesis of Parallel Algorithms*.

SET can be shown with **PolyTime** concepts.

Can it be shown with **NC²** concepts?

SET proves in **QLA** the following principles

1. $(\exists B \neq 0)[AB = I \vee AB = 0]$,
2. The columns of an $n \times (n + 1)$ matrix are linearly dependent,
3. Every matrix has an annihilating polynomial,
4. $AB = I \supset BA = I$,
5. **Existence of A^n , and the Cayley-Hamilton Thm.**

QLA can prove the existence of powers of a matrix from **SET**.

Let $\text{POW}(A, n)$ be the formula:

$$\exists \langle X_0 X_1 \dots X_n \rangle (\forall i \leq n) [X_0 = I \wedge (i < n \supset X_{i+1} = X_i * A)]$$

Show **QLA** $\vdash (\exists B \neq 0) [AB = I \vee AB = 0] \supset \text{POW}(A, n)$.

Let N be the $n^2 \times n^2$ matrix consisting of $n \times n$ blocks which are all zero except for $(n - 1)$ copies of A above the diagonal zero blocks^a.

^a*A taxonomy of problems with fast parallel algorithms*, by Stephen Cook.

Let N be the $n^2 \times n^2$ matrix consisting of $n \times n$ blocks which are all zero except for $(n - 1)$ copies of A above the diagonal zero blocks^a. Then $N^n = 0$, and $(I - N)^{-1} = I + N + N^2 + \dots + N^{n-1} =$

$$\begin{pmatrix} I & A & A^2 & \dots & A^{n-1} \\ 0 & I & A & \dots & A^{n-2} \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \end{pmatrix}.$$

Set $C = I - N$.

^a*A taxonomy of problems with fast parallel algorithms*, by Stephen Cook.

Let N be the $n^2 \times n^2$ matrix consisting of $n \times n$ blocks which are all zero except for $(n - 1)$ copies of A above the diagonal zero blocks^a. Then $N^n = 0$, and $(I - N)^{-1} = I + N + N^2 + \dots + N^{n-1} =$

$$\begin{pmatrix} I & A & A^2 & \dots & A^{n-1} \\ 0 & I & A & \dots & A^{n-2} \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \end{pmatrix}.$$

Set $C = I - N$.

Show that if $CB = 0$, then $B = 0$, using induction on the rows of B , starting with the bottom row.

^a*A taxonomy of problems with fast parallel algorithms*, by Stephen Cook.

Let N be the $n^2 \times n^2$ matrix consisting of $n \times n$ blocks which are all zero except for $(n - 1)$ copies of A above the diagonal zero blocks^a. Then $N^n = 0$, and $(I - N)^{-1} = I + N + N^2 + \dots + N^{n-1} =$

$$\begin{pmatrix} I & A & A^2 & \dots & A^{n-1} \\ 0 & I & A & \dots & A^{n-2} \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \end{pmatrix}.$$

Set $C = I - N$.

Show that if $CB = 0$, then $B = 0$, using induction on the rows of B , starting with the bottom row.

Using $(\exists B \neq 0)[CB = I \vee CB = 0]$, conclude that there is a B such that $CB = I$. Finally, show that $B = I + N + N^2 + \dots + N^{n-1}$.

^a*A taxonomy of problems with fast parallel algorithms*, by Stephen Cook.

Strong Linear Independence (SLI)

if $\{v_1, \dots, v_m\}$ are $n \times 1$, non-zero, linearly dependent vectors, then there exists a $1 \leq k < m$ such that

$$\underbrace{\{v_1, \dots, v_k\}}_{\text{lin. indep.}}, \underbrace{v_{k+1}, v_{k+2}, \dots, v_m}_{\text{lin. dep.}}$$

Csanky's algorithm (\mathbf{NC}^2) for computing the characteristic poly. of a matrix uses



symmetric polynomials:

$$s_0 = 1,$$

$$s_k = \frac{1}{k} \sum_{i=1}^k (-1)^{i-1} s_{k-i} \operatorname{tr}(A^i)$$

$$p_A(x) := s_0 x^n - s_1 x^{n-1} + s_2 x^{n-2} - \dots \pm s_n x^0.$$

Theorem: QLA proves the **Cayley-Hamilton Thm.** from SET and SLI.

The 12 steps proof :

(1) p_A is the characteristic polynomial of the matrix A as computed by Csanky's algorithm.

(2) Let $W = \{e_i, Ae_i, \dots, A^n e_i\}$.

(3) By SET, W must be linearly dependent.

(4) By SLI there exists a $k \leq n$ such that

$W_0 = \{e_i, Ae_i, \dots, A^{k-1} e_i\}$ is linearly independent and k is the largest such index.

(5) $A^k e_i$ can be written as a linear combination of the vectors in W_0 .

Let c_1, \dots, c_k be the coefficients of this linear combination, so that if $g(x) = x^k + c_1 x^{k-1} + \dots + c_k$, then $g(A)e_i = 0$.

(6) Let A_g be the $k \times k$ *companion matrix* of g ,

$$\left(\begin{array}{c|ccccc} 0 & 0 & 0 & \dots & 0 & -c_k \\ \hline 1 & 0 & 0 & \dots & 0 & -c_{k-1} \\ 0 & 1 & 0 & \dots & 0 & -c_{k-2} \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & -c_1 \end{array} \right)$$

(7) **LAP** proves $p_{A_g} = g$, and so **LAP** proves $(p_{A_g}(A))e_i = 0$.

(8) Extend W_0 to $B = W_0 \cup \{e_{j_1}, \dots, e_{j_{n-k}}\}$.

Existence of B follows from **SET**:

let $T = B_0 =$ the standard basis,

let $E = W_0$, which is linearly independent,

let $F = B_0 - \{e_{j_1}, \dots, e_{j_{n-k}}\}$,

so $B = (T - F) \cup E$.

$$A \sim \begin{pmatrix} A_g & E_1 \\ 0 & E_2 \end{pmatrix}$$

(9) **LAP** proves that if $C_1 \sim C_2$ then $p_{C_1}(x) = p_{C_2}(x)$
($\text{tr}(A) = \text{tr}(PAP^{-1})$, since $\text{tr}(AB) = \text{tr}(BA)$).

(10) **LAP** proves that if

$$C = \begin{pmatrix} C_1 & * \\ 0 & C_2 \end{pmatrix}$$

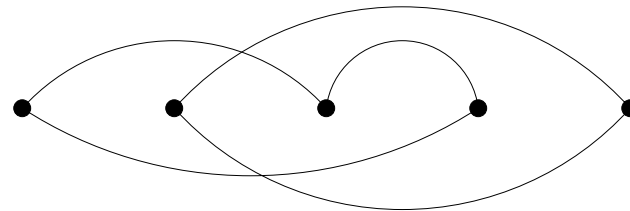
then $p_C(x) = p_{C_1}(x) \cdot p_{C_2}(x)$.

(11) $\therefore p_A(A)e_i = (p_{A_g}(A) \cdot p_E(A))e_i = p_E(A) \cdot (p_{A_g}(A)e_i) = 0$.

(12) This is true for all e_i in the standard basis, and so $p_A(A) = 0$.

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^n a_{i\sigma(i)}.$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 1 & 2 \end{pmatrix}$$



Cycle cover representation of permutations

A *clow*^a (*closed walk*) is a walk (w_1, \dots, w_l) starting from vertex w_1 and ending at the same vertex, where any (w_i, w_{i+1}) is an edge in the graph.

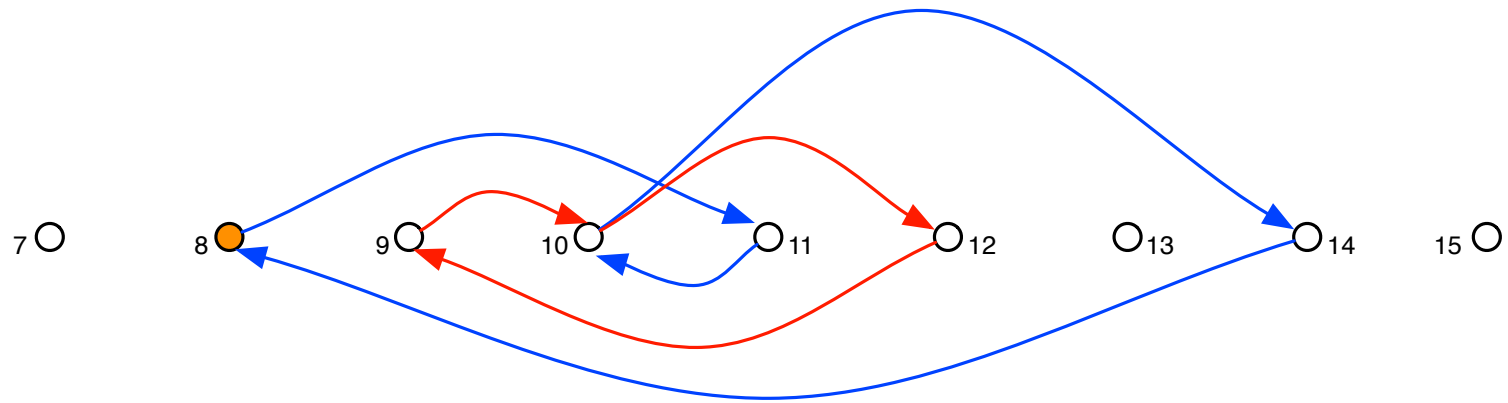
Vertex w_1 is the least-numbered vertex in the clow, and it is called the *head* of the clow. We also require that the head occur only once in the clow. This means that there is exactly one incoming edge (w_l, w_1) and one outgoing edge (w_1, w_2) at w_1 .

A *clow sequence* is a sequence of clows (C_1, \dots, C_k) with two properties: (i) the sequence is ordered by heads:

$$\text{head}(C_1) < \dots < \text{head}(C_k)$$

and (ii) the total number of edges, counted with multiplicity, adds to n .

^a*Determinant: Combinatorics, Algorithms, and Complexity*, by M. Mahajan and V. Vinay.



Clow (8, 11, 10, 12, 9, 10, 14)

$$\det(A) =$$

$$=$$
$$=$$

$$\det(A) = \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence}}} \text{sign}(\mathbf{C})w(\mathbf{C})$$

$$=$$
$$=$$

$$\begin{aligned} \det(A) &= \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence}}} \text{sign}(\mathbf{C})w(\mathbf{C}) \\ &= \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence} \\ \text{with head of 1st clow 1}}} \text{sgn}(\mathbf{C})w(\mathbf{C}) \\ &= \end{aligned}$$

$$\begin{aligned}\det(A) &= \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence}}} \text{sign}(\mathbf{C})w(\mathbf{C}) \\ &= \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence} \\ \text{with head of 1st clow 1}}} \text{sgn}(\mathbf{C})w(\mathbf{C}) \\ &= \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence} \\ \text{with } \textit{prefix} \text{ property}}} \text{sgn}(\mathbf{C})w(\mathbf{C})\end{aligned}$$

$$\begin{aligned}
 \det(A) &= \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence}}} \text{sign}(\mathbf{C})w(\mathbf{C}) \\
 &= \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence} \\ \text{with head of 1st clow 1}}} \text{sgn}(\mathbf{C})w(\mathbf{C}) \\
 &= \sum_{\substack{\mathbf{C} \text{ is a} \\ \text{clow sequence} \\ \text{with } \textit{prefix} \text{ property}}} \text{sgn}(\mathbf{C})w(\mathbf{C})
 \end{aligned}$$

$\mathbf{C} = (C_1, \dots, C_k)$ has the *prefix property* if $\forall i < k$ the total length of C_1, \dots, C_{i-1} is at least $\text{head}(C_i) - 1$.

A is $n \times n$.

Define layered, directed, acyclic graph H_A with three special vertices: s, t_+, t_-

such that

$$\det(A) = \sum_{\rho: s \rightsquigarrow t_+} w(\rho) - \sum_{\rho: s \rightsquigarrow t_-} w(\rho)$$

weight of path ρ is the product of weights of the edges appearing on it.

Main Idea: $s \rightsquigarrow t_+$ ($s \rightsquigarrow t_-$) paths will be in 1-1 correspondence with clow sequences of positive (negative) sign.

Vertices of H_A : $\{s, t_+, t_-\}$ together with

$$\{[p, h, u, i] : p \in \{0, 1\} \text{ (parity)}, h, u \in [n], 0 \leq i \leq n - 1\}$$

Meaning of the vertices: if a path ρ (starting from s) reaches a vertex $[p, h, u, i]$, this indicates that in the clow sequence being constructed along this path:

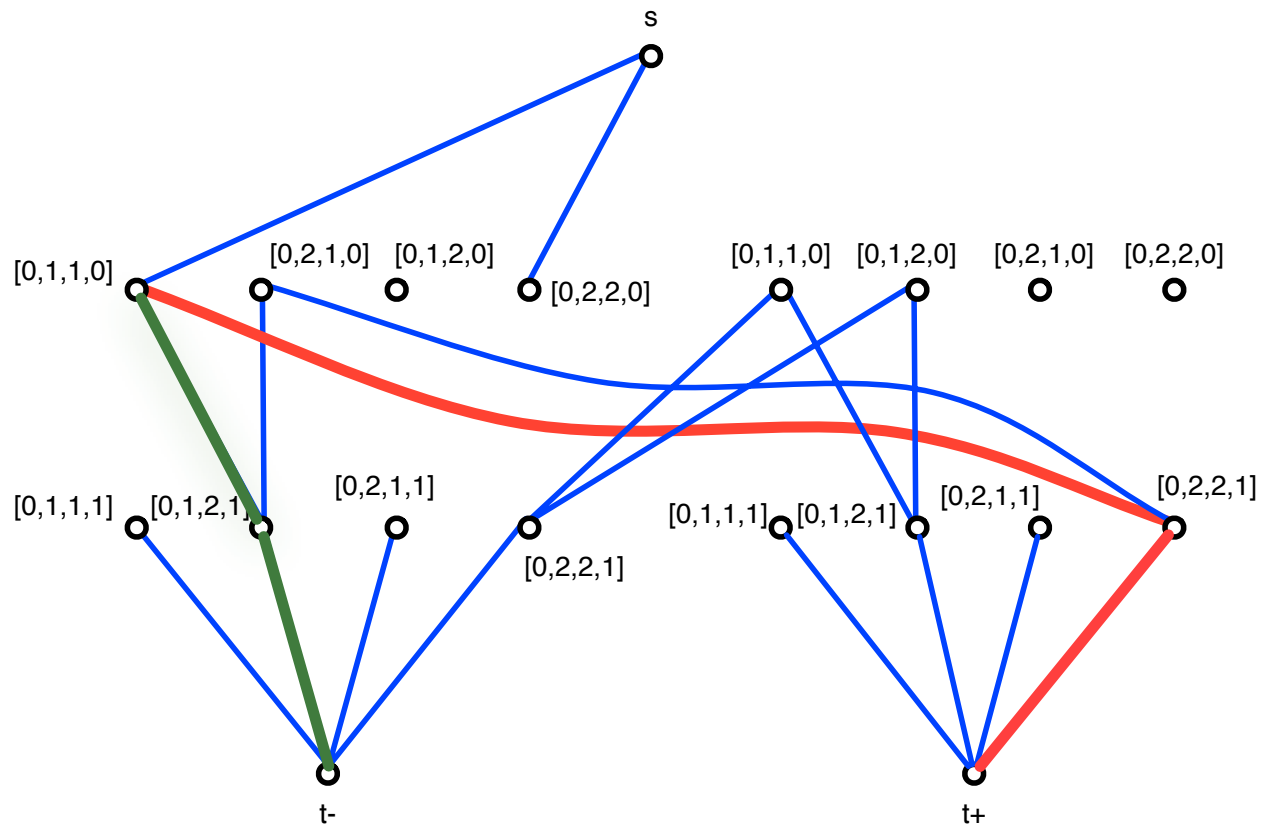
- p is the parity of the quantity “ $n+$ (the number of components already constructed),”
- h is the head of the clow currently being constructed,
- u is the vertex that the current clow has reached,
- and i is the number of edges traversed so far (in this and preceding clows).

Finally, a path from s to t_+ (t_-) corresponds to a clow sequence of positive (negative) parity.

The edges of H_A :

Edge	Condition	Weight
$(s, [b, h, h, 0])$	$h \in [n]$ and $b = \text{parity of } n$	1
$([p, h, u, i], [p, h, v, (i + 1)])$	$v > h$ and $(i + 1) < n$	a_{uv}
$([p, h, u, i], [\bar{p}, h', h', (i + 1)])$	$h' > h$ and $(i + 1) < n$	a_{uh}
$([1, h, u, (n - 1)], t_+)$		a_{uh}
$([0, h, u, (n - 1)], t_-)$		a_{uh}

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$



For $u, v, i \in [n]$ **and** $p \in \{0, 1\}$ (*Initialize values to 0*)

$$V[p, u, v, (i - 1)] \leftarrow 0$$

$V[t_+] \leftarrow 0$ **and** $V[t_-] \leftarrow 0$

$b \leftarrow$ **parity of** n

For $u \in [n]$ (*Set selected values at layer 0 to 1*)

$$V[b, u, u, 0] \leftarrow 1$$

For $i = 0$ **to** $(n - 2)$ (*Process outgoing edges from each layer*)

For $u, v \in [n]$ **such that** $u \leq v$ **and** $p \in \{0, 1\}$

For $w \in \{u + 1, \dots, n\}$

$$V[p, u, w, (i + 1)] \leftarrow V[p, u, w, (i + 1)] + V[p, u, v, i] \cdot a_{vw}$$

$$V[\bar{p}, w, w, (i + 1)] \leftarrow V[\bar{p}, w, w, (i + 1)] + V[p, u, v, i] \cdot a_{vu}$$

For $u, v \in [n]$ **such that** $u \leq v$ **and** $p \in \{0, 1\}$

$$V[t_+] \leftarrow V[t_+] + V[1, u, v, (n - 1)] \cdot a_{vu}$$

$$V[t_-] \leftarrow V[t_-] + V[0, u, v, (n - 1)] \cdot a_{vu}$$

Return $V[t_+] - V[t_-]$

- **Running time:** $O(n^4)$ -many edges in H_A , and one addition and one multiplication per edge.
- **Space:** at each stage only the values of two adjacent layers are required, so $O(n^2)$ entries; each N many bits.
- To compute $(p_A)_i$, we sum over clow sequences of $(n - i)$ many edges: add t_+^i, t_-^i , for each i , and connect t_+^i, t_-^i to the $(n - i)$ -th layer.
- If $V(t_+^i) - V(t_-^i) \neq 0$, then $i \leq \text{rank}_A$, and $\text{rank}_A \leq \text{rank}_G$, we can conclude that the matrix has rank at least i .

- **Pruning** of H_A : paths going through vertices of the form

$$[p, h, u, i] \quad \text{with } h > i + 1$$

cannot correspond to cycle covers (once h becomes a head, at least $(h - 1)$ edges should have been visited in the preceding cycle).

- **Prefix Property**: Extend the prefix property to *all* the coefficients,

$$(p_A)_i = (-1)^{n-i} \sum_{\substack{\mathbf{C} \text{ is an} \\ (n-i)\text{-clow sequence} \\ \text{with prefix property}}} \text{sgn}(\mathbf{C})w(\mathbf{C})$$

- Berkowitz's algorithm computes over clows with the prefix property ...