18 Sep 2023 Parallel Determinant Algorithm

<u>Announcement</u>. Office hours W 9/20 start at 4pm.

An <u>algebraic branching program</u> consists of:

(a) a directed acyclic graph (DAG) with vertices $v_1, \ldots, v_n$, edges (v_i, v_j) each satisfying $i < j$.

(b) a degree-1 multivariate polynomial $\ell(e)$ on each edge, in some set of variables $x_1, \ldots, x_m$.

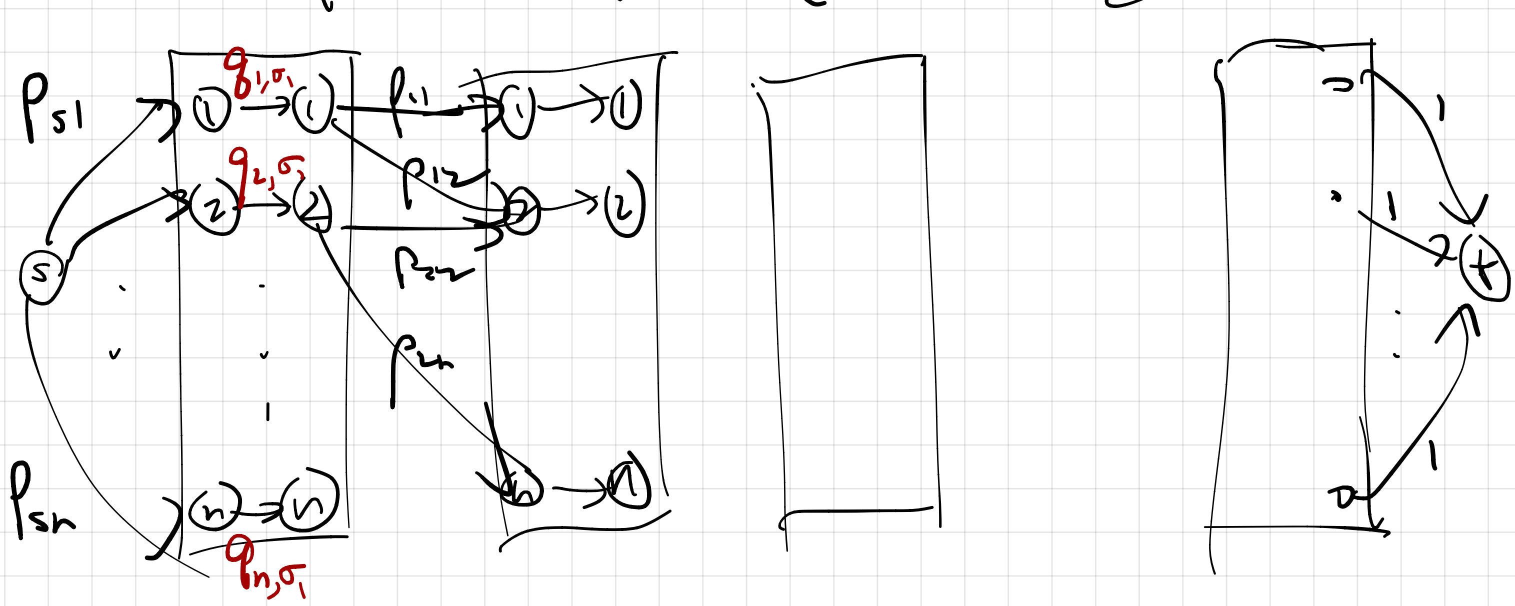
The polynomial computed by ABP, Π, is

$$f^{\Pi}(x_1, \ldots, x_m) := \sum_{\substack{\text{paths } P \\ v_1 \to v_n}} \prod_{e \in P} \ell(e)$$

<u>Example</u>. (probability of HMM producing string)

If we have HMM with state set S, starting state $s \in S$, alphabet Σ, transition probs p_{ij} $(i, j \in S)$, emission probs $q_{i\sigma}$ $(i \in S,\ \sigma \in \Sigma)$

the probability of producing string $\sigma_1, \sigma_2, \ldots, \sigma_n$ is $f^{\Pi}(p_{11}, p_{12}, \ldots, p_{nn}, q_{11}, \ldots, q_{nm})$ where Π is



A dynamic program to evaluate f^{π}:

define $\pi\langle j\rangle$ is the induced labeled subgraph of π on vertex set $\{v_1,\ldots,v_j\}$.

The recursive formula for $f^{\pi\langle j\rangle}$ is

$$f^{\pi\langle 1\rangle} = 1.$$

$$f^{\pi\langle j\rangle} = \sum_{e=(v_i,v_j)} \ell(e)\cdot f^{\pi\langle i\rangle}$$

For algebraic branching program π form matrix $M(\pi)$ where

$$M(\pi)_{ij} = \begin{cases} \ell(v_i,v_j) & \text{if } (v_i,v_j)\in E \\ 1 & \text{if } i=j=n \\ \emptyset & \text{otherwise} \end{cases}$$

Then f^{π} is the $(1,n)$ entry of $M(\pi)^{n-1}$.

To evaluate $f^{\pi}(a_1,\ldots,a_m)$ where $a_1,\ldots,a_m\in\mathbb{Z}$, substitute $a_1,\ldots,a_m$ for variables $x_1,\ldots,x_m$, obtain matrix $M=M(\pi;\vec{a})$, compute

$M^2, M^4, M^8, M^{16},\ldots, M^{2^k}$ ← largest power of 2 less than n

write $n-1$ in binary and the digits of $n-1$ tell you a subset of $\{M, M^2,\ldots,M^{2^k}\}$ whose product is M^{n-1}.

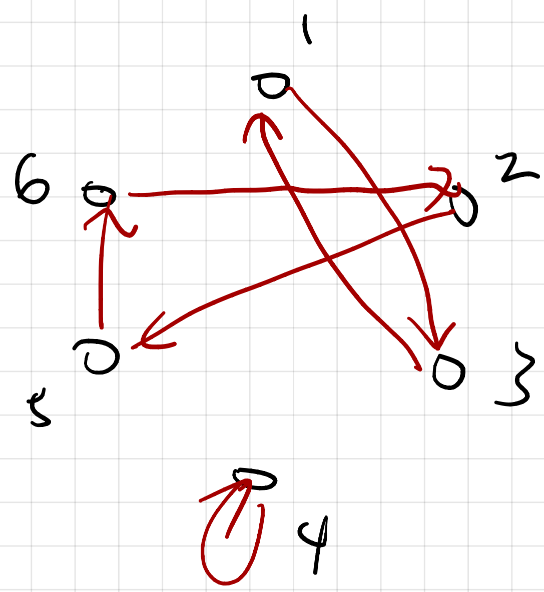
Evaluating f^{π} is $\le 2\log(n)$ mat mult's, so $O(\log^2 n)$ depth $O(\text{poly}(n))$ work.

Reducing determinant to evaluating poly(n)-size ABP.

Permutations as cycle diagrams.

E.g.

$1 \to 3$
$2 \to 5$
$3 \to 1$
$4 \to 4$
$5 \to 6$
$6 \to 2$



"closed walk"

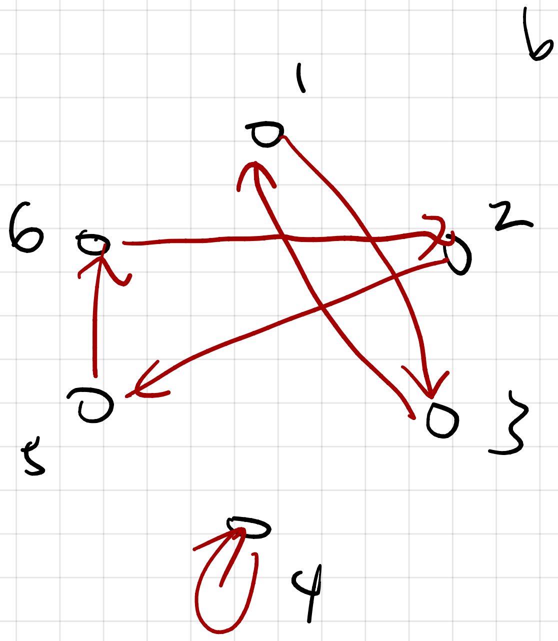
Def. A clow in a directed graph G is a sequence of vertices $C = v_0, v_1, v_2, \dots, v_k = v_0$ s.t. $(v_i, v_{i+1}) \in E(G)$ for all i.

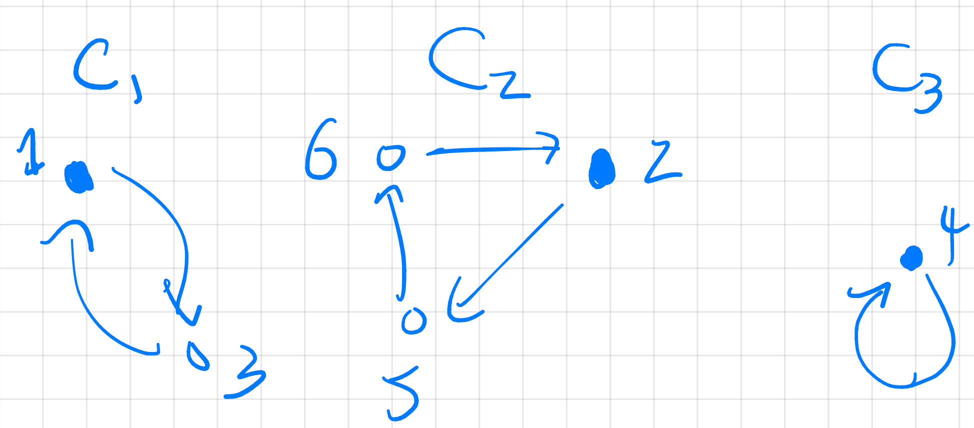
If G has edge labels, the weight of C is the product of its labels. The length of C is the # of edges.

If $V(G)$ is totally ordered, a clow sequence in G is a sequence of clows $C_1, C_2, \dots, C_k$ such that

a. The first and last vertex in each clow is its lowest-numbered vertex, called the "head"

b. The heads of clows are a strictly increasing sequence in $V(G)$.





The weight of a clow sequence is the product of the weights of the clows.

The length of a clow sequence is the sum of its lengths.

The sign of clow sequence $C_1, \ldots, C_k$ is $(-1)^{n-k}$

length

clows