Counting classes

- #SAT motivates the defn. of #P:

Defn: A fn. $f: \{0,1\}^* \to \mathbb{N}$ is in **#P** if there is a poly-time TM M & a constant $c > 0$ s.t. $\forall x$, $f(x) = \#\{y \in \{0,1\}^{|x|^c} \mid M(x,y)=1\}$.

$\Rightarrow$ #P is the collection of functions that **count** the # accepting-paths of an efficient NDTM.

Proposition: (i) $\#SAT \in \#P$.

(ii) $FP \subseteq \#P$. Open: $FP \neq \#P$?

(iii) $NP \subseteq P^{\#P} \subseteq Pspace$.

$P^{\#P}$: Using a fn. as an oracle.

- #P has an analogous decision version:

Defn: A lang. $L \in$ **PP** if $\exists$ poly-time TM M & a constant $c > 0$ s.t. $\forall x$, $x \in L$ iff $\#\{y \in \{0,1\}^{|x|^c} \mid M(x,y)=1\} \geq \frac{1}{2} \cdot 2^{|x|^c}$.

prob. poly-time

- A function in PP computes the most-significant-bit of the corresponding fn. in #P !

Proposition: (1) $\#SAT \in FP^{PP}$.
(2) $P^{\#P} = P^{PP}$.

Proof:

(1) Let ϕ be a boolean formula in vars. $x_1, \ldots, x_n$.

- Using PP we could know whether $\#\phi := \#(\text{sat.-assgn. of } \phi) \geq 2^{n-1}$ or not.
- We'll instead use PP to find the k_{min} → minimum k st. $k + \#(\phi) \geq 2^n$.

We'll use binary search & n+1 non-det. bits.

- Consider an NDTM M that on input $\phi(\bar{x})$ & k, uses non-det. bits $y_0, \ldots, y_n$ as follows:

i) On $y_0 = 1$, M guesses $\bar{y} = (y_1, \ldots, y_n)$ & outputs $\phi(\bar{y})$.

ii) On $y_0 = 0$, M accepts exactly k strings $\bar{y} \in \{0,1\}^n$.

- Clearly, $\#\text{acc-path}(M,(\phi,k)) = k + \#(\phi)$
& possible paths are 2^{n+1} many.
$\Rightarrow$ $k + \#(\phi) \geq 2^n$ can be learnt by querying the PP-oracle on (ϕ, k).
- We can find k_{min} (from msb to lsb) bit-by-bit, each using a query to PP-oracle.

$\Rightarrow$ Using n calls to the oracle we find k_{min} & hence we compute $\#(\phi)$.

$\Rightarrow$ $\#SAT \in FP^{PP}$.

(2). • We have shown $P^{\#SAT} \subseteq P^{PP}$.
• Later we will show that #SAT is #P-complete.
$\Rightarrow$ $P^{\#P} \subseteq P^{PP}$.
• The other direction is trivial. □

#P-completeness

- We call a fn. $f: \{0,1\}^* \to \mathbb{N}$ #P-complete if $f \in \#P$ & $\#P \subseteq FP^f$.

Turing reduction

functions computable by a poly-time TM using f as an oracle.

▷ If an $f \in FP$ is #P-complete, then $FP = \#P$.

Theorem: #SAT is #P-complete.

Proof:
- We know that $\#SAT \in \#P$.
- Since the computation of a poly-time NDTM can be encoded in a boolean formula, we deduce:

For any $f \in \#P$ with $f(x) = \#$(accepting paths of an NDTM M on x), we get a formula $\varphi_{M,x}$ s.t. $f(x) = \#$(sat. assgn. of $\varphi_{M,x}$).

Cook-Levin reduction is parsimonious

$\Rightarrow f \in FP^{\#SAT}$

$\Rightarrow \#P \subseteq FP^{\#SAT}$. □

An algebraic #P-complete problem

- For a matrix $A \in \mathbb{F}^{n \times n}$, <u>permanent</u> is defined as:

$$\underline{per(A)} := \sum_{\sigma \in Symm_n} \prod_{i=1}^{n} A_{i,\sigma(i)}$$

$Symm_n$ is the group of $n!$ permutations of $[n]$.

- Notice the similarity with

$$det(A) = \sum_{\sigma \in Symm_n} sgn(\sigma) \cdot \prod_{i \in [n]} A_{i,\sigma(i)}.$$

- We will see that $per(\cdot)$ is one of the hardest problems in #P!

<u>Lemma 1:</u> Per, for 0/1 matrices, is in #P.

<u>Proof:</u> • Let $A \in \{0,1\}^{n \times n}$.

• $per(A) = \#\{\sigma \in Symm_n \mid \prod_{i=1}^{n} A_{i,\sigma(i)} = 1\}$.

$\Rightarrow$ $per(A)$ is the #(acc. paths of the NDTM that given A guesses the σ). $\square$

- Permanent has several known applications in computational physics & probability.

Graph interpretation of per

Defn:
- Given $A \in \mathbb{F}^{n \times n}$, view it as the adjacency matrix of a weighted digraph on n vertices.
- A cycle-cover C of A is a subgraph having n vertices, each vertex of in-deg = out-deg = 1.
- Thus, C is a union of disjoint cycles.
- $wt(C)$:= product of weights of edges in C.

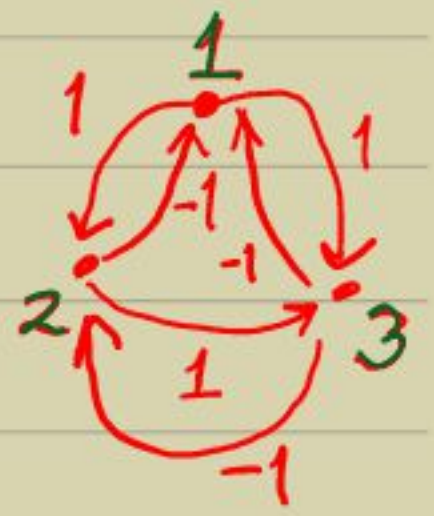
▷ $$per(A) = \sum_{C \in \text{cycle-cover}(A)} wt(C) .$$

Pf. sketch: • Say, a monomial $A_{11}A_{23}A_{32}A_{44}$ appears nontrivially in per(A).

⇒ The digraph G, corr. to A, has the cycle-cover:

• Conversely, it can be seen that the wt. of a cycle-cover of G corresponds to a monomial of per(A). □

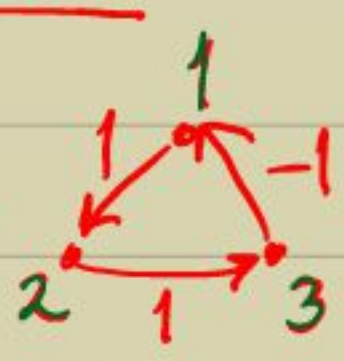
Eg. $A = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}$; G =



Cycle-covers of A:

wt = 1

&



wt = −1

; per(A) = 1 − 1 = 0.

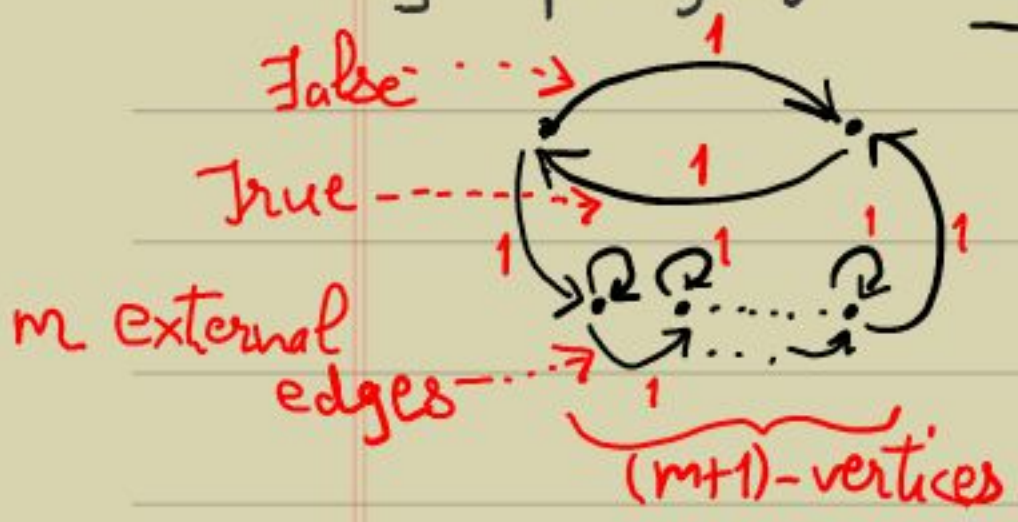
Theorem: per for 0/1 matrices is #P-hard.

Proof:

- We will relate #3SAT to permanent.
- Let φ be a 3CNF formula with m clauses & n variables.

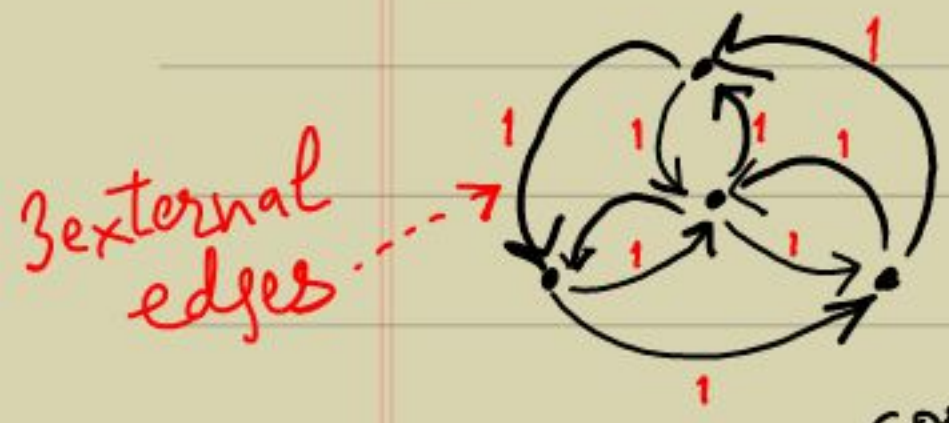
  Wlog each clause has exactly three literals.
- We will construct a graph A s.t. the sum of its weighted cycle-covers $=: \text{per}(A) = \#(\text{sat. assgns. of } \varphi)$.

- Each variable x_i is converted to a graph gadget V_i:



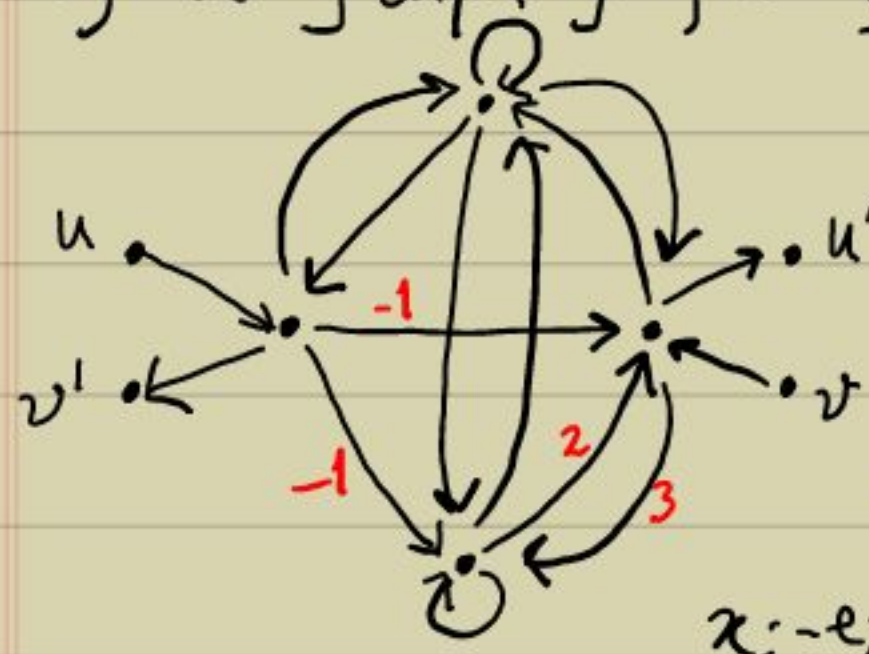
- The cycle-cover using the external edges signifies $x_i = T$ while the other signifies $x_i = F$.

• Each clause F_j is converted to a graph gadget $\underline{C_j}$:



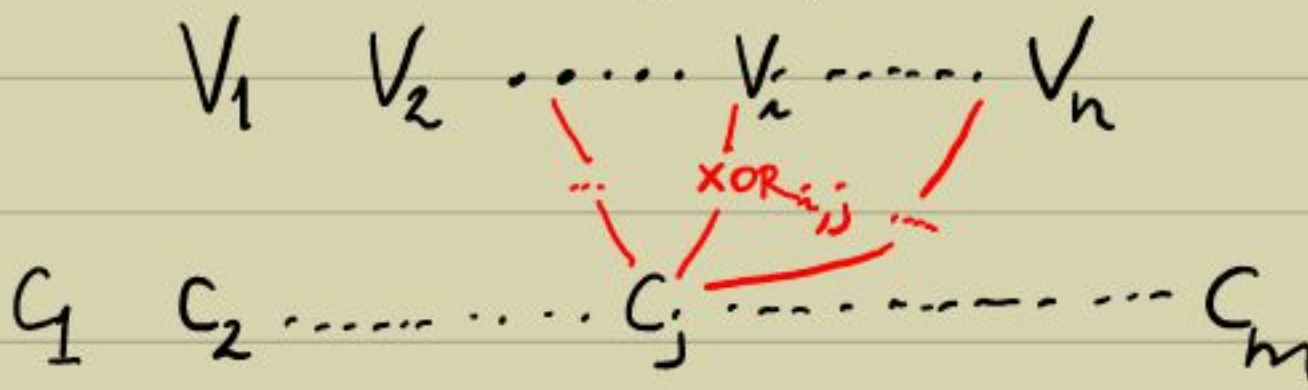
• There are 3 cycle-covers each of wt=1, each corresponding to a dropped external edge. The latter signifies that the corresponding literal in F_j is True.

• If x_i is in F_j then the j-th external edge (u, u') of V_i is connected to the x_i-external-edge (v, v') of C_j by a graph gadget $\underline{XOR_{i,j}}$:



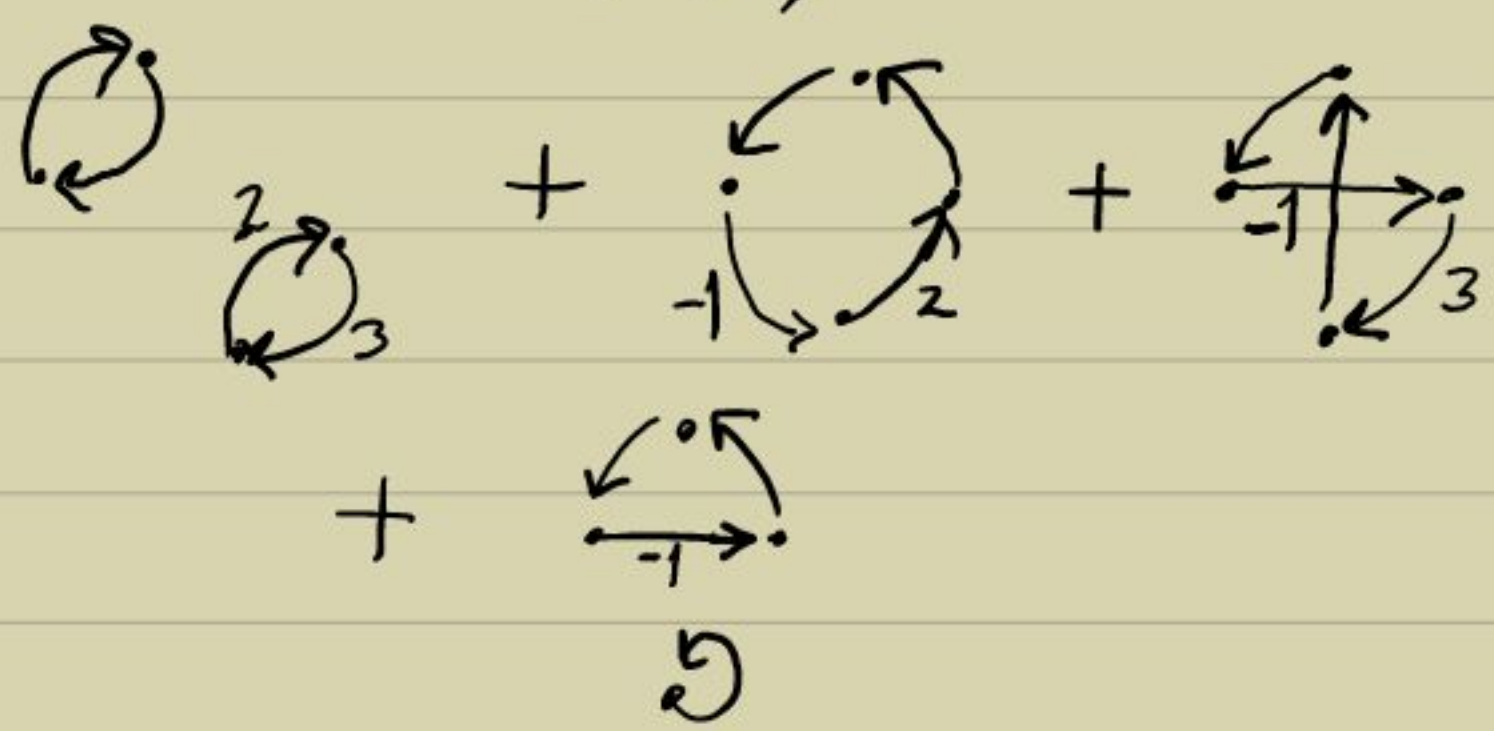
• If $\bar{x}_i$ is in F_j then the False edge (u, u') of V_i is connected to x_i-external-edge (v, v') of C_j by $XOR_{i,j}$.

• Finally, the graph A' is:



<u>Claim:</u> The cycle covers of XOR sum to weight 0.

<u>Pf:</u> • Cycle-covers in the gadget $XOR \setminus \{u, u', v, v'\}$:



$= 6 - 2 - 3 - 1 = 0.$ □

• Thus, the cycle-covers of A' involving $XOR_{i,j}$ are those that correspond to <u>exactly</u> one of the paths (u, u') or (v, v').

• Consider such a cycle-cover $\mathcal{C}$ of A':

For every C_j, there is an external edge, say x_i, not appearing in $\mathcal{C}$.

$XOR_{i,j}$ ensures that the j-th external edge in V_i appears in $\mathcal{C}$.

$\Rightarrow \mathcal{C}$ uses the external cycle-cover of V_i

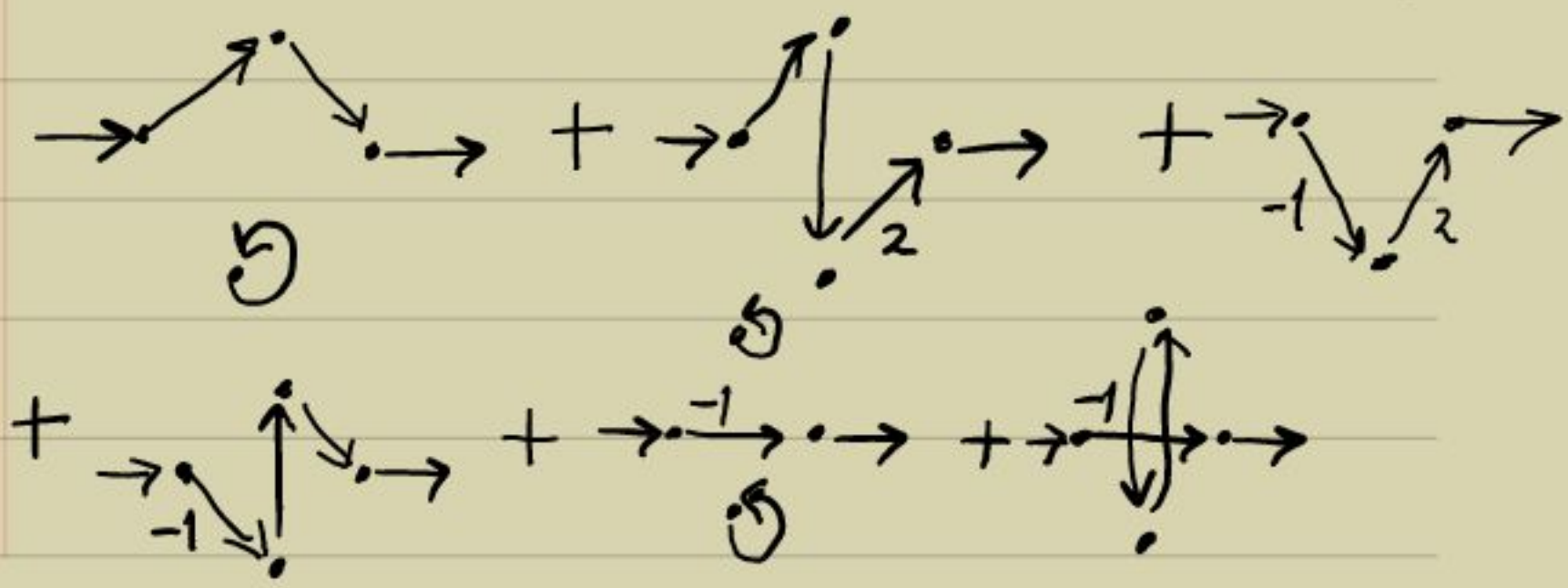
$\Rightarrow x_i = \text{True}$ is captured!

$\Rightarrow \mathcal{C}$ corr. to a sat. assgn. of ϕ.

(The converse follows by the gadget defn.)

Claim: The contribution of $XOR_{i,j} \setminus \{v, v'\}$ to the weight is $-2 \neq 0$. (* char. of the field not 2.)

Pf: • Paths from u to u':



$= 1+2-2-1-1-1 = -2.$ □

$\Rightarrow$ For a satisfying assignment s of ϕ: $\sum_{\substack{\text{cycle-cover } C \\ \text{of } A' \text{ corr. to } s}} wt(C) = (-2)^{3m}$.

the #clauses is m & each has exactly 3 literals $\Rightarrow$ #XOR's $= 3m$.

$\Rightarrow per(A') = \sum_{\text{cycle-cover } C \text{ of } A'} wt(C)$

$= (-2)^{3m} \cdot \#(\text{sat. assgns. of } \phi)$.

- Finally, we want to reduce A' to a 0/1 matrix.

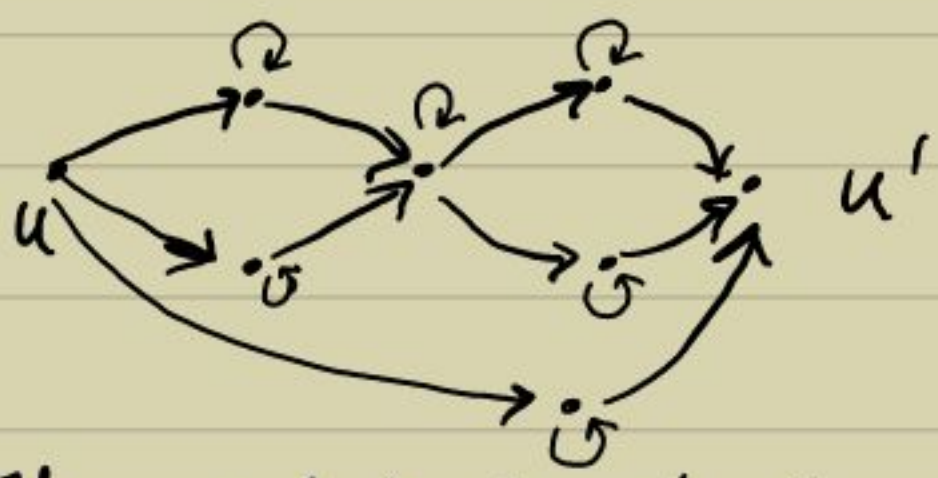
(i) $|per(A')| < 2^{3m} \cdot 2^{n+1} =: 2^N$.

So, we can replace a negative entry $-k$ in A' by $(2^N - k)$; get a matrix A'' & consider $per(A'') \pmod{2^N}$.

(ii) A'' has non-negative, but large, entries.

Replace a weighted edge $u \xrightarrow{k} u'$ by a gadget of <u>sequential</u> & <u>parallel</u> weight ≤ 2 paths.

Eg. $u \xrightarrow{5} u'$ by:



$5 = 2^2 + 2^0$

- This gadget is of size $O(\lg^2 k)$.

▷ Finally, we get a 0/1 matrix A, of dim. $\text{poly}(m+n)$ s.t.

$$\text{per}(A) \bmod 2^N = (-2)^{3n} \cdot \#\Phi .$$

$\Rightarrow \#SAT \in FP^{\text{per on 0/1}}$

$\Rightarrow \#P \subseteq FP^{\text{per on } 0/1}$. $\square$

Theorem (Valiant 1979): per, on 0/1 matrices, is #P-complete.