CPSC 536F Feb 15, 2022

Next week is break week.

=

After Valiant's permanent paper,

I'll mention some more results

in alg. comp. thy

Then: graphs, eigenvalues, ...

=

Valiant's gadgets:

Valiant's gadgets:

$$X = \begin{pmatrix} 0 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 3 & 0 \end{pmatrix}$$

has: $\operatorname{Perm} X = 0$,

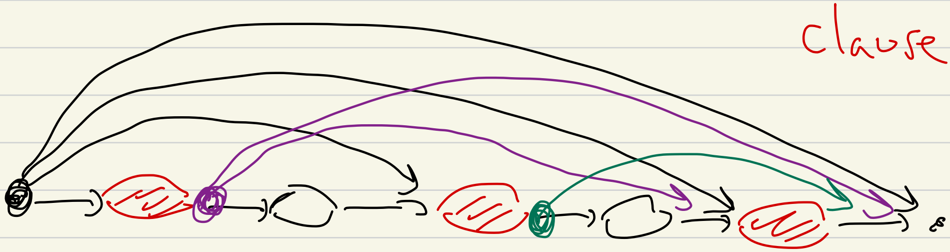
$$\operatorname{Perm} X(1;1) = \operatorname{Perm} X(4;4)$$

$$= \operatorname{Perm} X(1,4;1,4) = 0$$

and

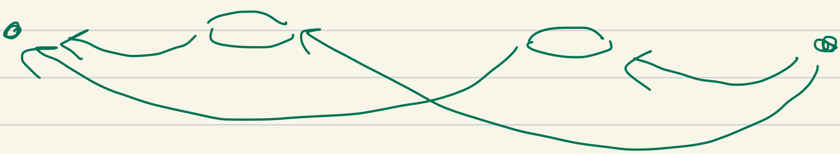
$$\operatorname{Perm} X(1;4) = \operatorname{Perm} X(4;1) = 4$$

each clause

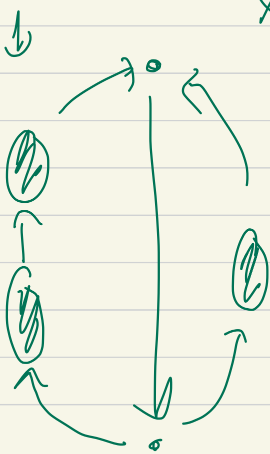


plus downward arrows

Interchange



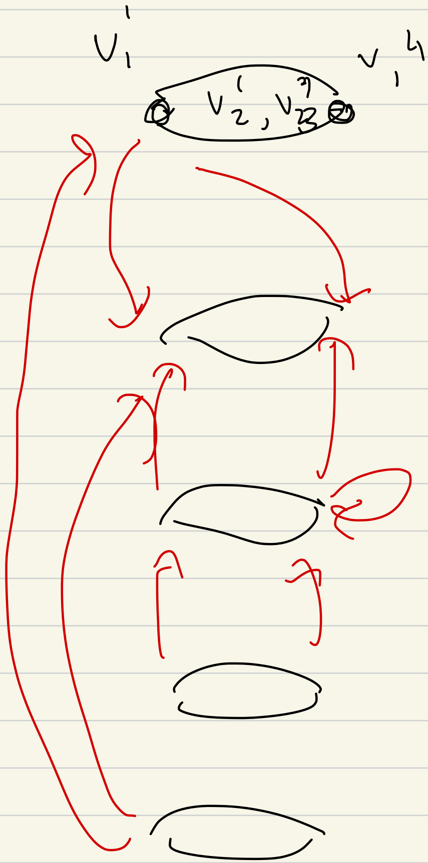
x_i $\bar{x}_i$



each variable

"track"

Point: Say you have a graph, made up of



$\longrightarrow$ extra edges.

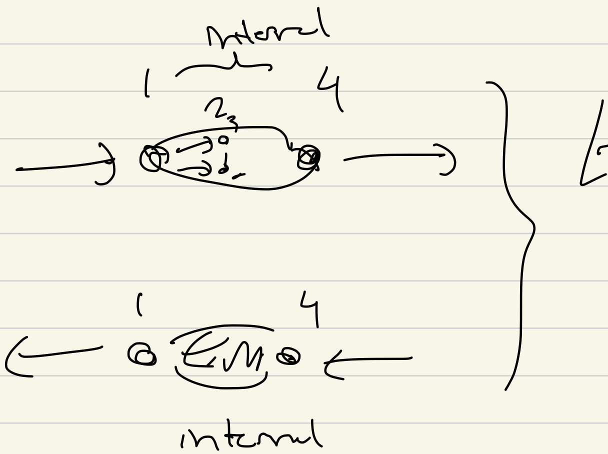
Any edge not shown

$i \underbrace{\dashrightarrow}_{\text{no edge}} j \Rightarrow X_{ij} = 0$

Valiant's 4x4 gadget

Claim:

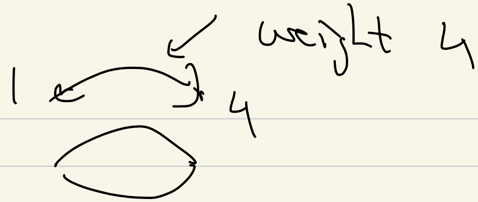
Route = a set of cycles in big graph s.t.



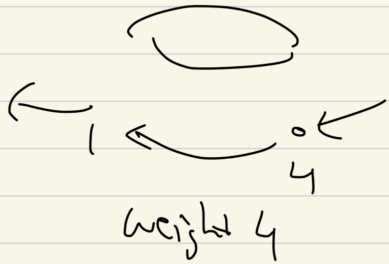
+ other edges

then the only non-zero contrib to permanent is

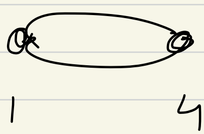
for each Valiant piece, and each such piece can be replaced by

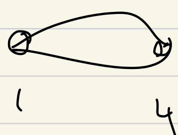


or



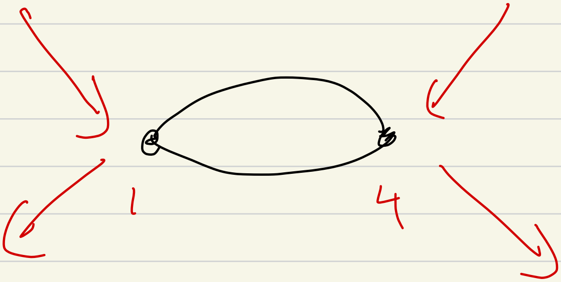
Idea: consider all union of cycles in $V=\{v_1, \dots, v_m\}$, and for each Valiant piece, look at





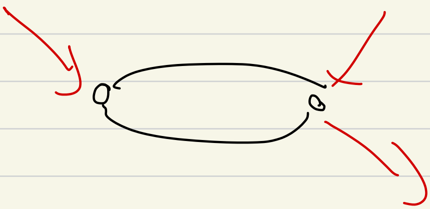
the 2,3 vertices aren't connected

to exterior nodes, so look at red edges :



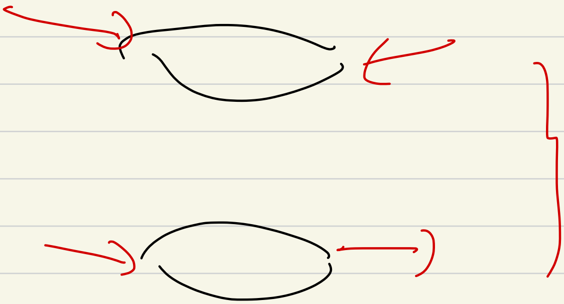
2 edges incident upon v_1

2 edges .. v_4

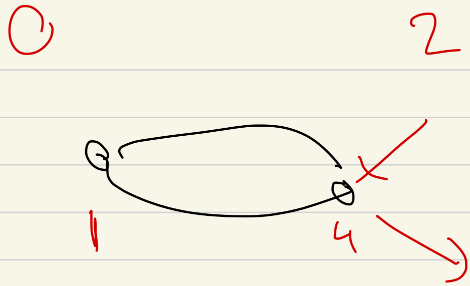


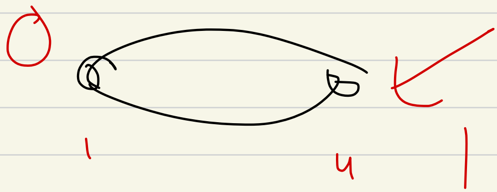
1 edge, the vertex 1

2 edges vertex 4



1 and 1

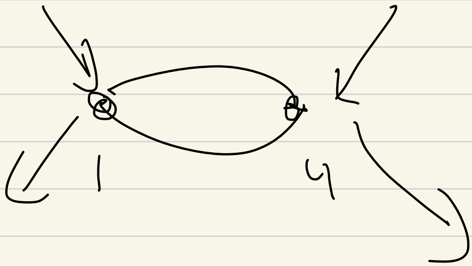




Sum the permanent!

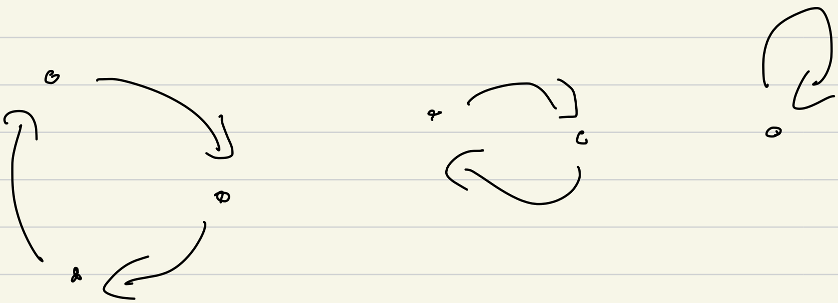
$$\sum_{\sigma : \{v_1, \dots, v_m\} \to \{v_1, \dots, v_m\}}$$

subdivide σ's into their behaviour at each e.g.



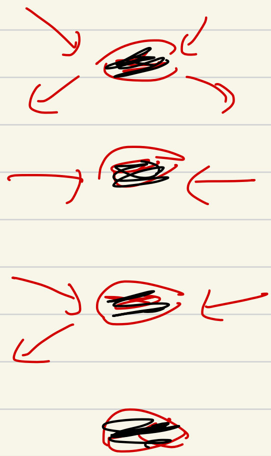
possible!

cycle decomp of σ



directed graph, each vertex has indegree 1, outdegree 1

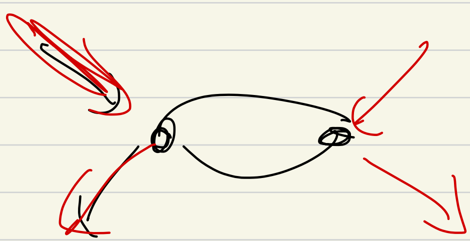
Graph



other red edges

etc.

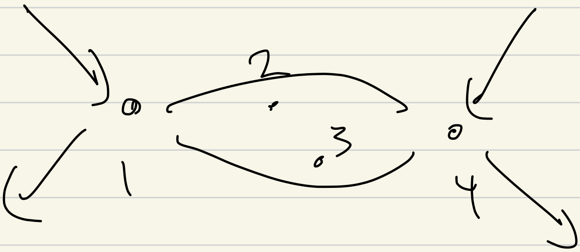
Consider cycle that on some



Contrib to perm

has to sum to 0

Since



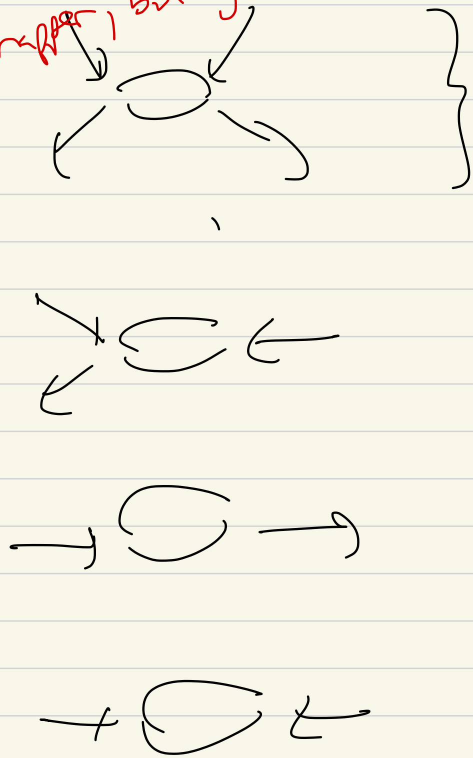
So $2, 3$ are taken to themselves under σ, sum over all poss

$$X = \begin{array}{c c} & \begin{array}{cccc} & 2 & 3 & \end{array} \\ \begin{array}{c} \\ 2 \\ 3 \\ \\ \end{array} & \begin{pmatrix} 0 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 3 & 0 \end{pmatrix} \end{array}$$

$$\sum_{\substack{\text{such} \\ \sigma}} \operatorname{Perm} \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} = 0$$

$\sigma : v_2^i, v_3^i \to v_2^i, v_3^i$, and - - -

Can happen, but gives 0 contrib. to Perm



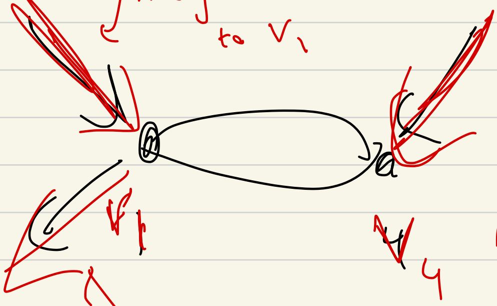
any such situation gives all σ with this pattern summing to 0 in Perm X

$$X \in \{-1, 0, 1, 2, 3\}^{m \times m}$$

$m = \#$ vertices

What about:

indegree 1 to v_1



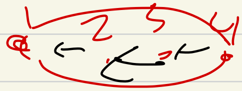
or

? external edges ?

outdegree 1 to v_1

Rem

indegree 1 to v_4
outdegree 0 to v_4

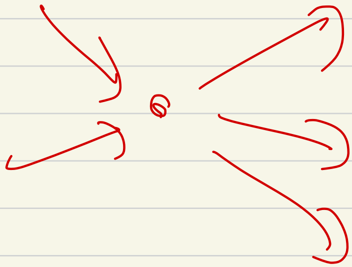


each internal edge

has indegree = 1

outdegree = 1

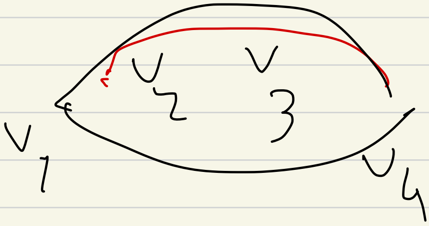
So sum over all indegree edges in vert 1,2,3,4 = all outdegree



recall! on a digraph, the

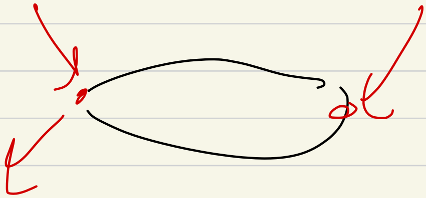
indegree(v) = # edges pointing to v

outdegree(v) = # edges pointing away from v



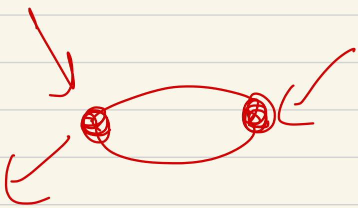
$$\left.\begin{array}{l}\text{the total indegree} \\ = \text{the total outdegree}\end{array}\right\} \text{internal edges}$$

hence



total external indegree = total external outdegree

$\Rightarrow$



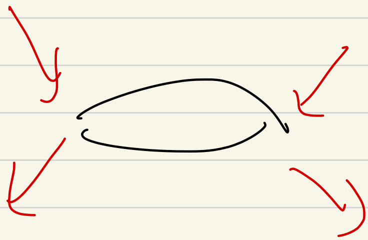
Impossible!

external outdegree = 1

external indegree = 2.

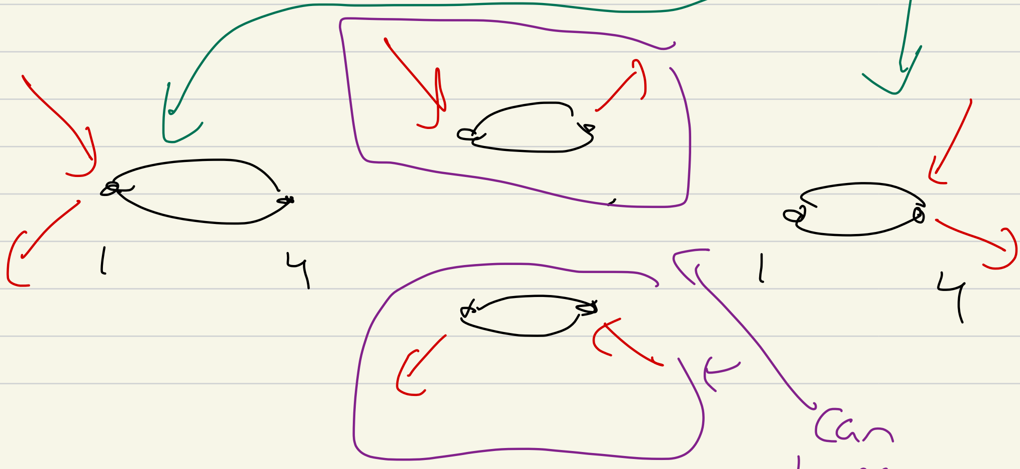
For graph theoretic reasons!

(1) external indegree
" outdegree = 2



anything with this structure ⇒ 0 contribs to perm

(2) external indegree
external outdegree } = 1

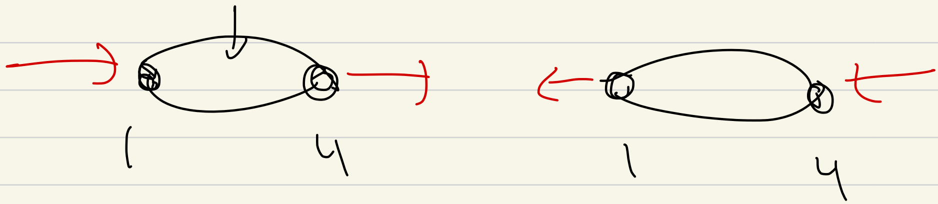


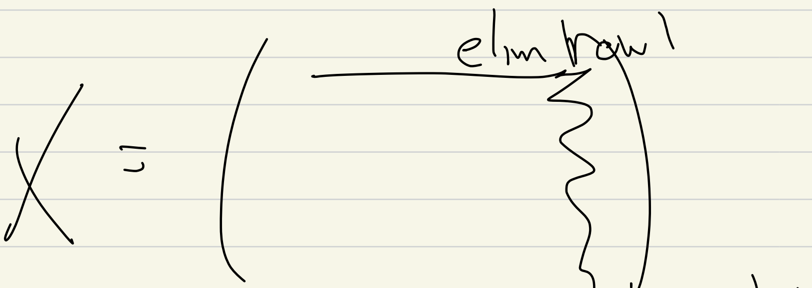
can happen

Since

$$\mathrm{Perm}\left(X\left(\begin{matrix}\text{elim}\\ \text{row } 1\end{matrix}\ ;\ \begin{matrix}\text{elim}\\ \text{col } 1\end{matrix}\right)\right) = 0$$

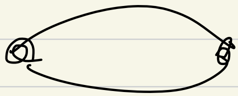
all contrib from internal part





$$\mathrm{Perm}\left(X(1;4)\right) = \mathrm{Perm}\left(X(4;1)\right) = 4$$

last case

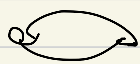
3 

external indegree = " outdegree

no external edges = 0

$$\mathrm{Perm}(X) = 0$$

the sum over all such $\sigma : \bar{V} \to \bar{V}$

with this behavior at 

is 0.

By a case analysis, and

$$\operatorname{Perm}(X) = \operatorname{Perm}(X(1,4;1,4))$$

$$= \operatorname{Perm}(X(1;1))$$

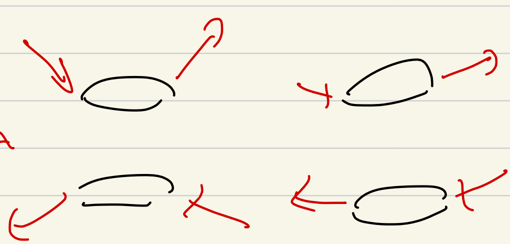
$$= \operatorname{Perm}(X(4;4)) = 0$$

but

$$\operatorname{Perm}(X(1;4)) = \operatorname{Perm}(X(1;4))$$

$$= \text{const, here } = 4$$

every $\operatorname{Perm}(X)$:



each contrib $4^{\#\text{ of Valiant pieces}}$

Next!

Say you have

f

$(x_1 \text{ OR } x_2 \text{ OR } \neg x_3)$ AND

$(x_5 \text{ or } \neg x_{12} \text{ or } x_4)$ AND

⋮

$(\qquad\qquad)$

Build a digraph with edge weights

so that # satisfying assignments

$= \left(4 \leftarrow \text{sum of \# literals and clauses}\right)$

$\cdot$ (# satisfying assignments)

Digraph graph size is

poly (size of f)

=

Break 10:18 – 10:23

Have f = 3CNF formula ;

Say that

$$x_1 = T, \quad x_2 = F, \quad x_3 = F, \ldots$$

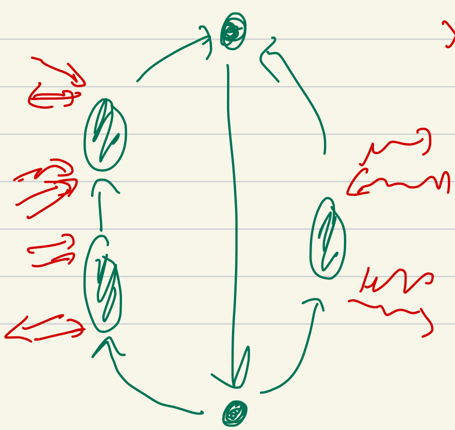
satisfies $f(x_1, \ldots, x_n) = T$:

$\Rightarrow$

for $i = 1, \ldots, n$, part of the graph we build

only edges in out top vertex

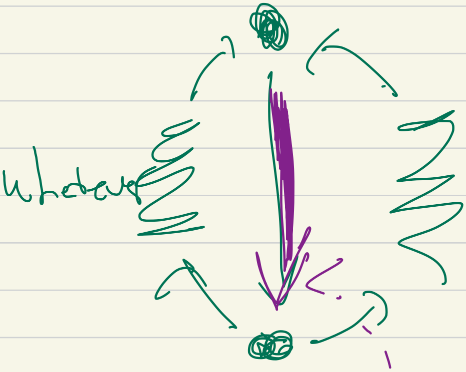
x_i



there will be stuff

only edges in/out bottom vertex

Cycle must have indegree = outdegree = 1 at each vertex

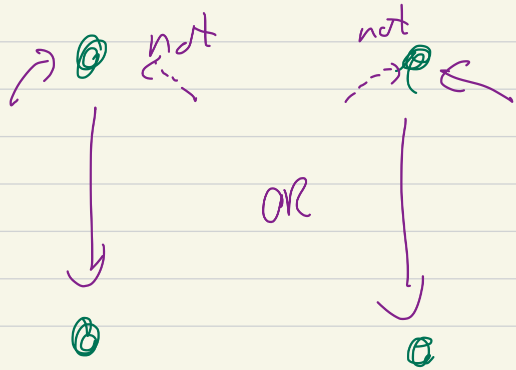


whatever

must be involved in cyclic decomp of

$\sigma : V \to V$

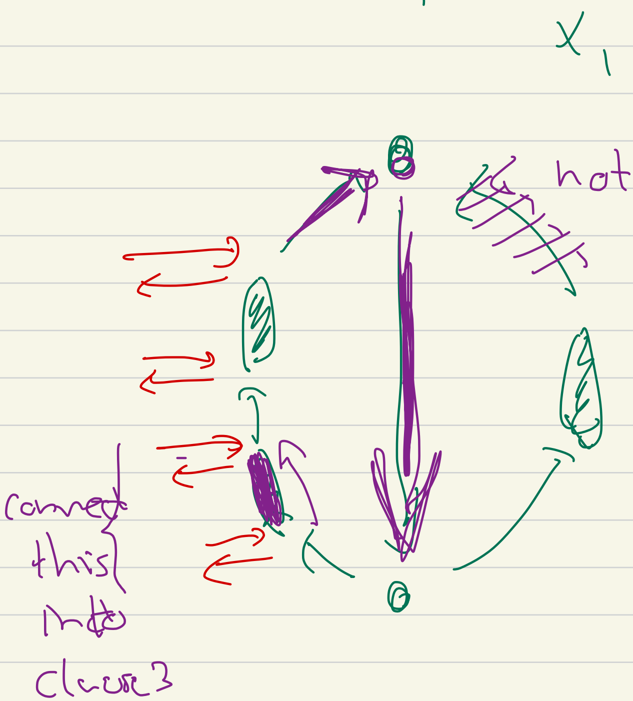
For x_i



x_i = true

x_i = false

More precisely



say that top/bottom "track"

x_1 occurs in formula:

x_1 occurs twice

$\neg x_1$ occurs once

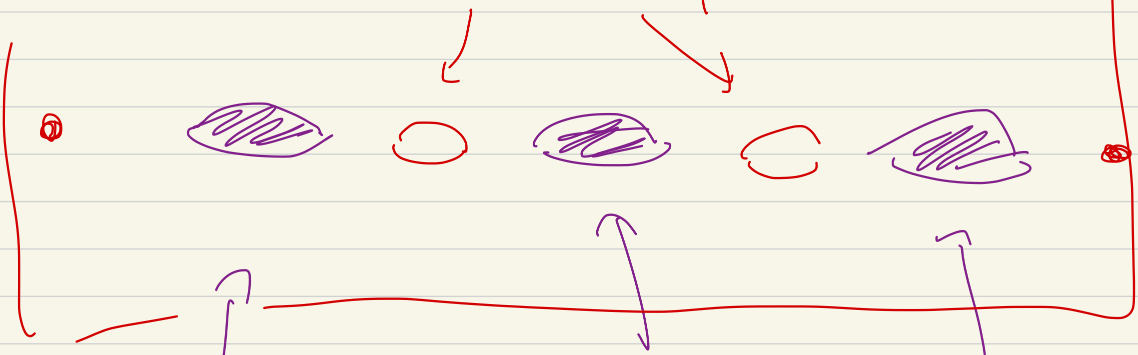
Clauses

clause 3

$$(\ x_1 \ \text{OR} \ x_2 \ \text{OR} \ \neg x_5\ ) \quad \text{AND} \quad \vdots \quad \text{AND} \quad \vdots$$

~~"interchange"~~

auxilliary Valiant pieces

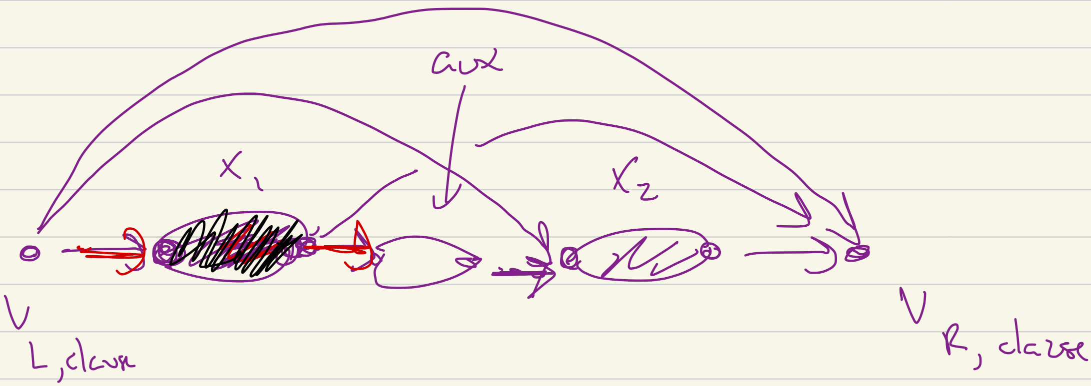


from x_1 side (left) of $x_1, \bar{x}_1$ track

from x_2 (left) side of track x_2

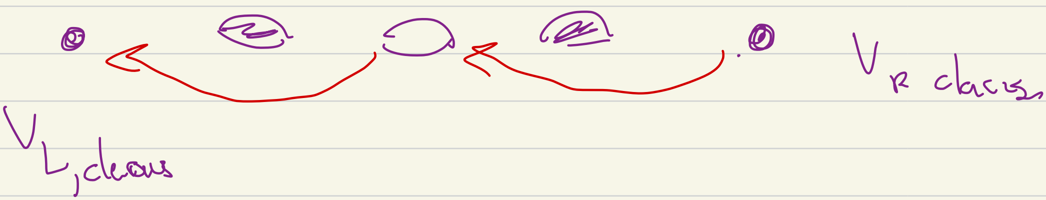
from $\neg x_5$ (right) side of track x_5

Let's do x_1 or x_2 2CNF clause



left to right part

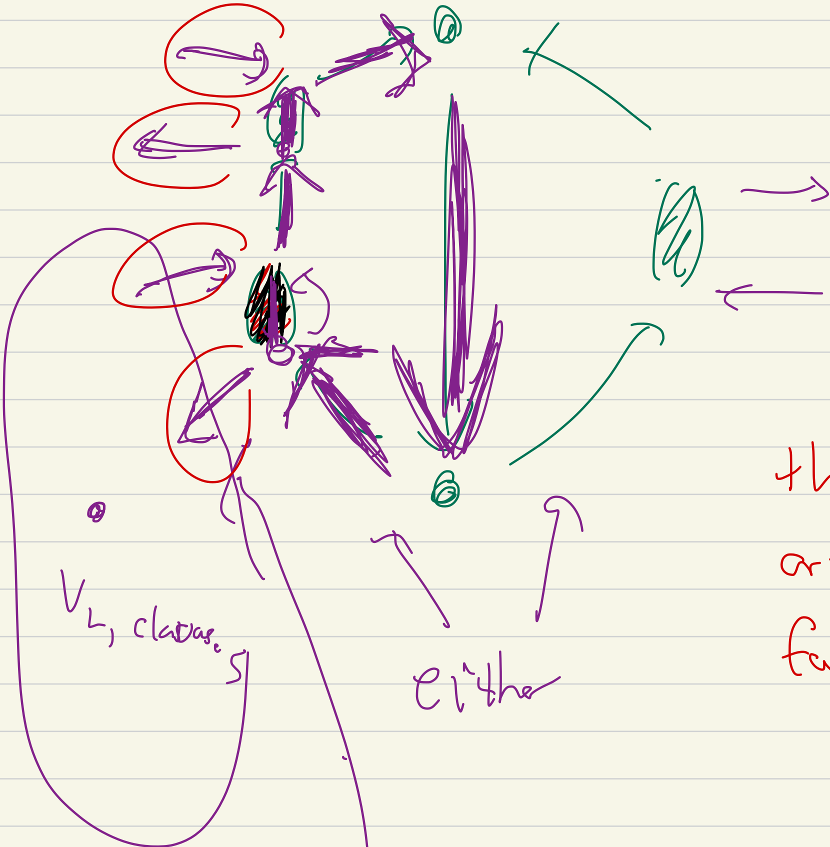
add edges



you must traverse at least one piece

track for x_i!

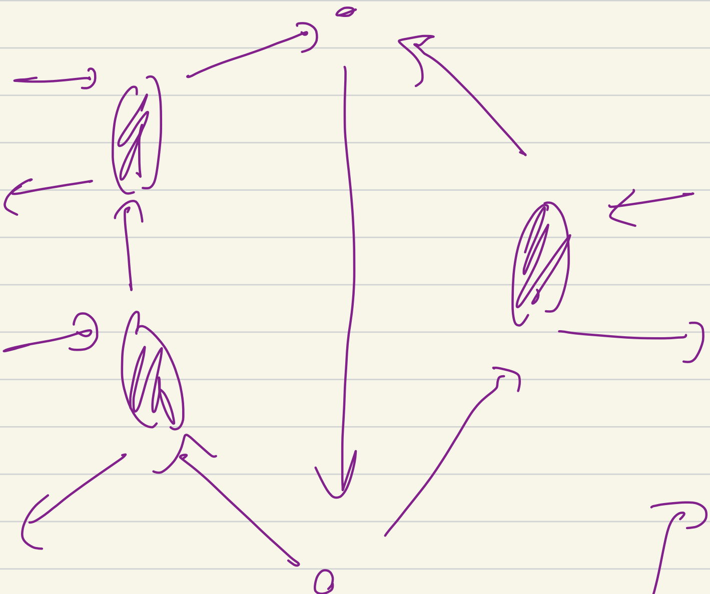
claim:



this orientation forces

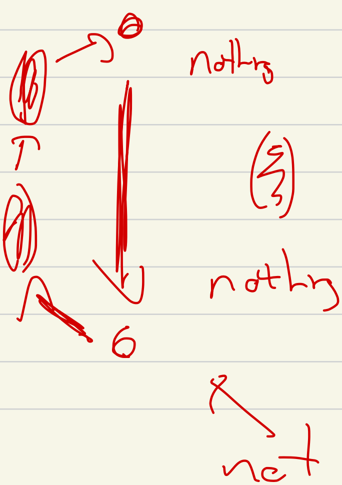
can't do

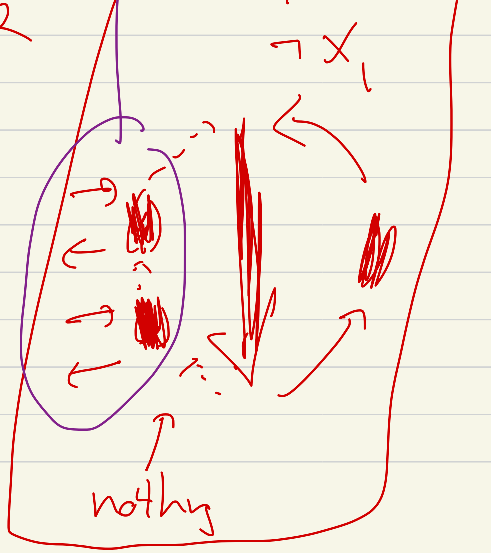
So either



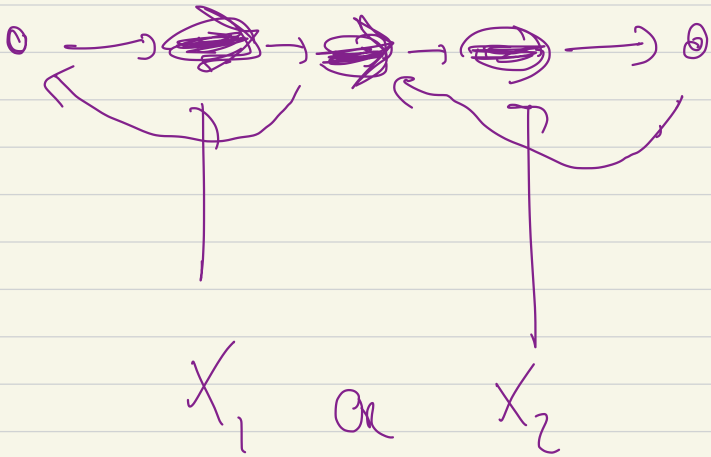
each gets indegree from clause

OR

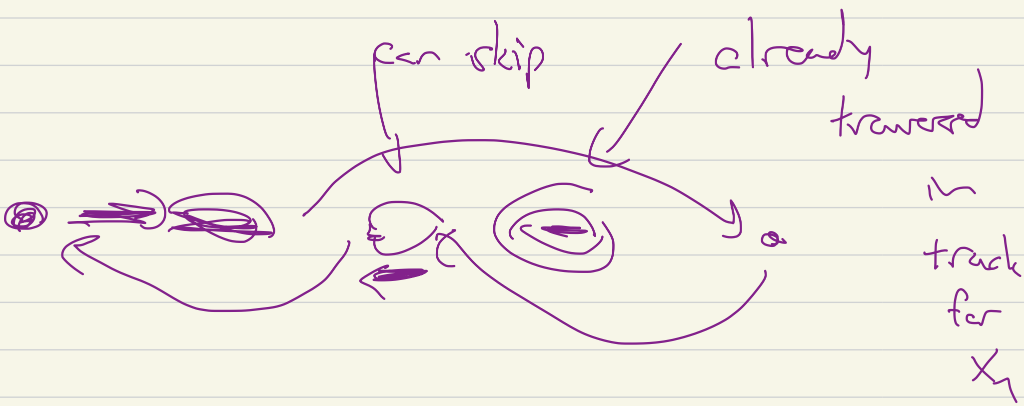




if $\neg x_1$ chosen
and $\neg x_2$ chosen



if $\neg x_1$ chosen, x_2 chosen



Class ends.