# Lecture 15

Last lecture we saw

1. Modern SAT solvers are successful in practice
2. Running time and space can be modelled in Proof Complexity
3. Small space and short length simultaneously

implies that there are GOOD PROTOCOLS for communication problems

* We came out with a CNF such that the communication protocol is hard.

## Main Theorem

There are k-CNF formulas $\{F_n\}_{i=1}^{\infty}$ of size $O(n)$ such that

- resolution can refute $F_n$ in length $O(n)$
- any PC and CP refutation of $F_n$ with

length $L$ and space $s$

have

$$s \log L \geq \sqrt[4]{n}$$

(2)

# PROOF of the MAIN THEOREM

(A) $P_n$ = Pebbling formulas on Pyramid graphs

(B) $F_n = Lift_3(P_n)$

↳ $F_n$ has resolution refutation of length $O(n)$

(C) In PC and CP it holds that

$$s \log L \geq bs_{crit}(Search(P_n)) \quad \text{[Lemma 1]}$$

(D) $bs_{crit}(Search(P_n)) = \Omega(\sqrt{n})$ [Corollary 4]

---

**Today** we prove (A), (B), (D) and we discuss

Search(Lift(F)) vs Lift(Search(F))

---

## Lift of a formula in CNF form

Let $F$ be a $k$-CNF $\bigwedge_i^m C_i$ on $n$ variables $V_1, \dots, V_n$.

**$Lift_\ell(F)$** is a formula on

- $2n\ell$ variables
- $\binom{\ell}{2} n + m\ell^k$ clauses

  $\binom{\ell}{2} n$ → OPTIONAL
- $Lift_\ell(F)$ is a $\max\{2k, \ell\}$-CNF

| ORIGINAL VARIABLES | X variables | Y variables |
|---|---|---|
| $v_1$ | $x_{1,1}\ x_{1,2} \cdots x_{1,\ell}$ | $y_{1,1}\ y_{1,2} \cdots y_{1,\ell}$ |
| $v_2$ | $x_{2,1}\ x_{2,2} \cdots x_{2,\ell}$ | $y_{2,1}\ y_{2,2} \cdots y_{2,\ell}$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $v_n$ | $x_{n,1}\ x_{n,2} \cdots x_{n,\ell}$ | $y_{n,1}\ y_{n,2} \cdots y_{n,\ell}$ |

the INTENDED MEANING is that

Y variables encode a selector which activates one of the X variables in the same block to be the input of the original CNF.

" at least one variable X in each block must be selected "

$$\forall\, i \in [n] \qquad y_{i,1} \vee y_{i,2} \vee y_{i,3} \cdots y_{i,\ell}$$

" at most one variable in each block must be selected "

$$\forall_{i \in [n]}\ \forall_{a,b \in \binom{[\ell]}{2}} \qquad \overline{y_{i,a}} \vee \overline{y_{i,b}}$$

[this last set of clauses is optional]

These clauses on Y variables allow to consider a one-to-one correspondence with input of the

(4)

Now we want to encode the clauses: consider an arbitrary clause $C_i$ and without loss of generality assume that

$$C_i = \underbrace{V_1 \text{ OR } V_2 \text{ OR } \cdots V_r}_{\text{positive literals}} \text{ OR } \underbrace{\overline{V_{r+1}} \text{ OR } \cdots \text{ OR } \overline{V_s}}_{\text{negative literals}}$$

For every map $f : [s] \to [l]$ we add the following clause $C_i^f$

$$\left[ Y_{1,f(1)} \text{ AND } Y_{2,f(2)} \text{ AND } \cdots Y_{s,f(s)} \right]$$

$$\Rightarrow$$

$$\left[ X_{1,f(1)} \text{ OR } X_{2,f(2)} \text{ OR } \cdots X_{r,f(r)} \text{ OR } \overline{X_{r+1,f(r+1)}} \text{ OR } \cdots \right.$$
$$\left. \cdots \overline{X_{s,f(s)}} \right]$$

which in clause form can be written as

$$\bigvee_{i=1}^{r} \left( \overline{Y_{i,f(i)}} \vee X_{i,f(i)} \right) \quad \vee \quad \bigvee_{i=r+1}^{s} \left( \overline{Y_{i,f(i)}} \vee \overline{X_{i,f(i)}} \right)$$

Any clause $C_i$ with $s$ literals is encoded as a set of $l^s$ clauses of $2s$ literals

# Some observation about $Lift_\ell(F)$

$$Lift_\ell(F) \in UNSAT \iff F \in UNSAT$$

this is clear since any assignment to X and Y variables encodes an assignment for the original variables and any such assignment has many encodings in X and Y variables.

Pick $\vec{v} \in \{0,1\}^n$ and $\vec{x}$ and $\vec{y} \in \{0,1\}^{n\ell}$ encoding the same assignment for F.

- If a clause $C_i \in F$ is not satisfied, then focus on $C_i^f \in Lift_\ell(F)$

observe $f$ is the mapping implicitly defined where

without loss of generality

$$C_i = V_1 \vee \cdots \vee V_r \vee \bar{V}_{r+1} \vee \cdots \vee \bar{V}_s$$

Let $f(j) = t$ where $Y_{j,t} = 1$

then $C_i^f$ is falsified.

- If $\left(\overline{Y_{1,t_1}} \vee \cdots \vee \overline{Y_{s,t_s}}\right) \vee \left(X_{1,t_1} \vee X_{2,t_2} \vee \cdots \vee \overline{X_{s,t_s}}\right)$

is false then and $C_i = V_1 \vee V_2 \vee \cdots \vee \bar{V}_s$

$V_j = X_{j,t_j}$ is false

- For assignments $X$ and $Y$ which exactly encode assignment for $\vec{V}$ variables we get

$$Lift_\ell(F) \text{ false} \leftrightarrow F \text{ false}$$

- For assignments to $X$ and $Y$ which does not encode $\vec{V}$ assignments, initial clauses in $Lift_\ell(F)$ are falsified.

---

Structure of the clauses in $Lift_\ell(F)$

Any $C_i^P = C_Y \vee C_X$ where $C_Y$ is on $Y$-vars and $C_X$ is on $X$-vars

Given an assignments on $\vec{X}$ and $\vec{Y}$, a

$C_Y$ is false only if $Y$ induces the corresponding mapping

If $\vec{Y}$ encodes a mapping $[n] \to [\ell]$ then

$$Lift_\ell(F)|_Y \cong F$$

## About the optional clauses

if we exclude (as it is done in the paper) the clauses $\boxed{\overline{Y_{j,a}} \vee \overline{Y_{j,b}}}$

then $\vec{Y}$ encodes a multifunction and

$Lift_\ell(F)|_Y$ contains different overlapping copies of formulas isomorphic to $F$.

So even in this case $\boxed{Lift_\ell(F) \in UNSAT \iff F \in UNSAT}$

---

In the rest of the lecture we do not consider these optional clauses, since it is not done even in the paper.

The main point is that the assignment that we are going to consider are such that

$\boxed{Y \text{ encodes a proper function}}$

So from the point of view of search problems their presence is indifferent.

(8) Claim

Short and small space refutation for

$$Lift_\ell(F)$$

gives a low communication protocol for

$$Lift_\ell(Search(F))$$

---

- Now we review the proof that efficient proofs give efficient protocols

- from that proof we get an efficient protocol for

$$Search(Lift_\ell(F))$$ such that

- it works for every division of the variables between Alice and Bob
- it works also for assignments that are not encoding an assignment for $Vars(F)$.

It is easy now that if Alice and Bob needs to solve an instance of $Lift_\ell(Search(F))$ they can use such protocol

Alice get $Y = [Y_1, \dots, Y_n] \in [n]^\ell$ and encodes it as a bit matrix $Y = \begin{bmatrix} Y_{1,1} & \cdots & Y_{n,1} \\ Y_{1,2} & & \\ \vdots & & \end{bmatrix}$ where $Y_{i,Y_i} = 1$ and $0$ otherwise

Bob gets $X = \begin{bmatrix} x_{1,1} & \cdots & x_{n,1} \\ \vdots & & \vdots \\ x_{1,\ell} & \cdots & x_{n,\ell} \end{bmatrix}$ and leaves the input as it is.

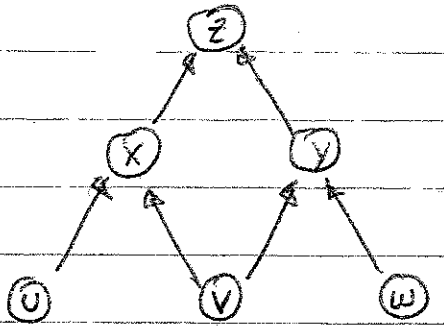
they use the protocol for Search($Lift_\ell(F)$) and they collectively compute a clause in $Lift_\ell(F)$ which is false.

Since $Y$ encodes a mapping $[n] \to [\ell]$ then the falsified clause must be of the form $C_i^{F}$

which means that $C_i$ is the answer to

$Lift_\ell(Search(F))$ □

## ⑩ Pebbling Formulas on Pyramids




$P_n$

$u$ , $v$ , $w$

$\bar{u} \vee \bar{v} \vee x$ (i.e $u \wedge v \rightarrow x$)

$\bar{v} \vee \bar{w} \vee y$ (i.e $v \wedge w \rightarrow y$)

$\bar{x} \vee \bar{y} \vee z$ (i.e $x \wedge y \rightarrow z$)

$\bar{z}$

- It has a refutation of length $O(n)$ by unit propagation.

---

Now we want to show that

$F_n = Lift_3(P_n)$ has a refutation of length $O(n)$.

We leave this as an exercise.

For the rest of the lecture we focus on proving that

$$bs_{crit}(Search(P_n)) \geq \Omega(\sqrt{n})$$

## Block Sensitivity

- of a function $f: \{0,1\}^n \to A$ on assignment $\alpha$

Let $B \subseteq [n]$ and let $\alpha^B = \begin{cases} \alpha_i \text{ if } i \notin B \\ \neg\alpha_i \text{ if } i \in B \end{cases}$

$bs(f,\alpha)$ = max # of blocks $B_1 \dots B_t$ such that

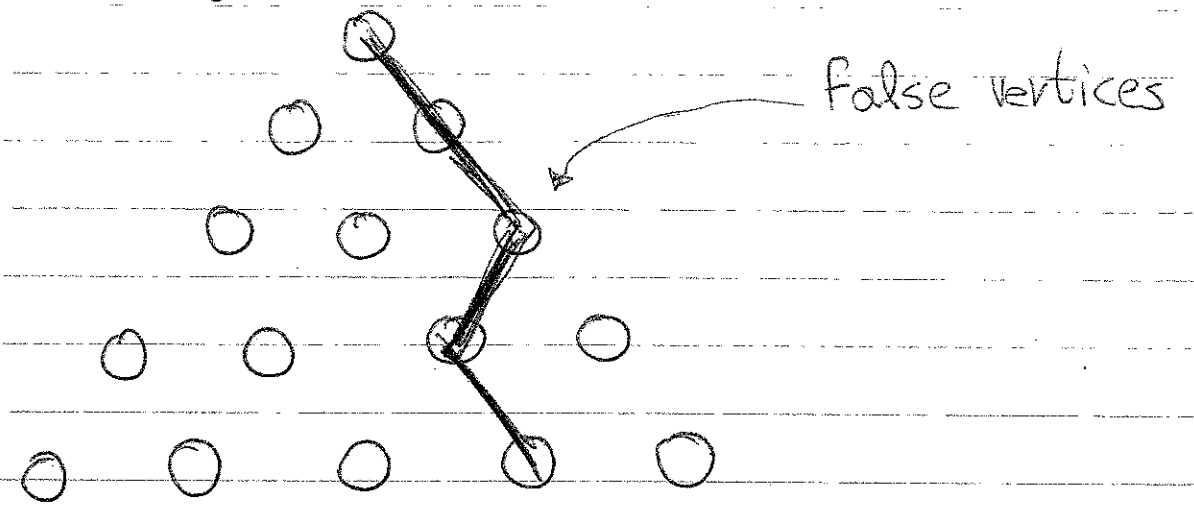
- $B_i \cap B_j = \emptyset \quad \forall i \neq j$
- $f(\alpha^{B_i}) = f(\alpha) \quad \forall i$

Critical Block Sensitivity for $S \subseteq \{0,1\}^n \times A$

- Let $A_{crit} = \{q \mid \#\{(q,a) \in S\} = 1\}$
- $f$ <u>solves</u> $S$ if $\forall q \in \{0,1\}^n \quad (q, f(q)) \in S$

- $bs_{crit}(S) = \max_{\alpha \in A_{crit}} \min_{f \text{ solves } S} bs(f,\alpha)$

(12) Interesting critical assignments




We consider paths from source to sink

- the only falsified clause corresponds to the source node

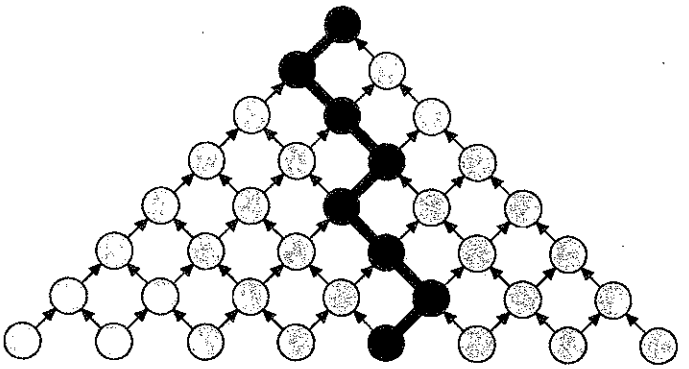
**PATH GRAPH**

We define the following structure:

$$G = (V, E)$$

$$V = \{ \text{all path from a source to the sink} \}$$

# Critical Assignments for Pyramid Pebbling Contradiction



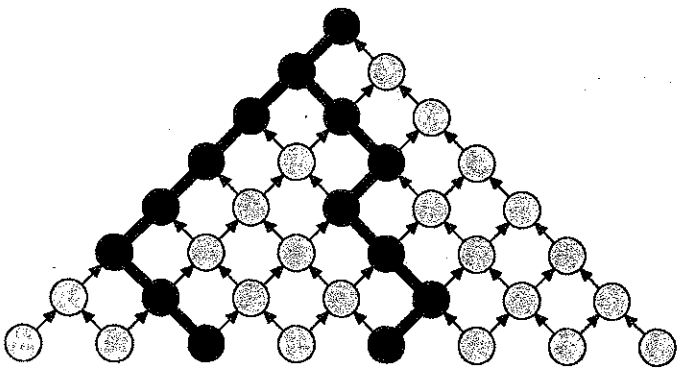
Focus on critical assignment setting:

- vertices on one source-to-sink path $P$ false
- all other vertices true (so source($P$) only correct answer)





# Critical Assignments for Pyramid Pebbling Contradiction



Focus on critical assignment setting:

- vertices on one source-to-sink path $P$ false
- all other vertices true (so source($P$) only correct answer)

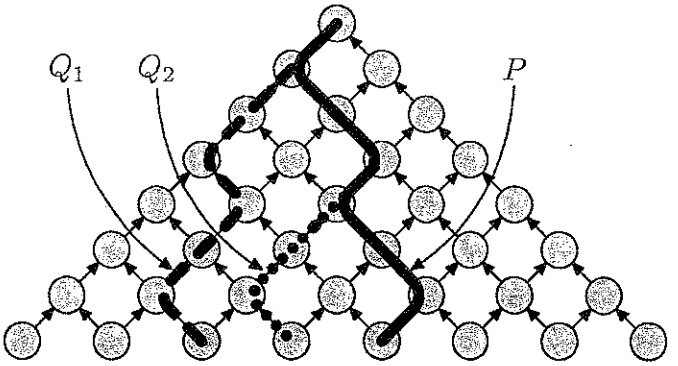
Bicritical assignments falsify two different paths

$\Rightarrow$ two possible correct answers

(14)



## Path Graph




Build graph $G$ such that

- vertices = source-to-sink paths $P$
- edge $(P, Q)$ only if $P$ and $Q$ merge and stay together
- in addition, if $(P, Q_1)$ and $(P, Q_2)$ edges, then $Q_1 \cap Q_2 \subseteq P$
- $G$ is undirected — $(P, Q)$ edge only if $(Q, P)$ edge



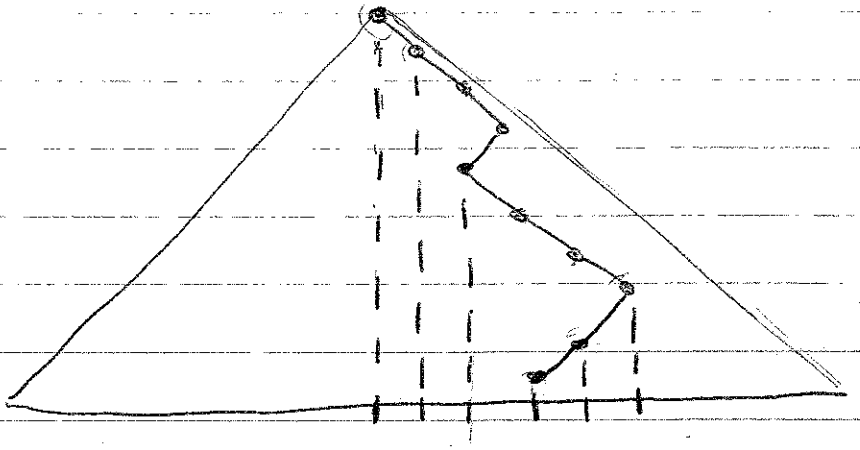


## Dense Path Graph ⇒ High Critical Block Sensitivity

**Lemma 2**

If $\exists$ path graph $G$ with average degree $d$, then falsified clause search problem for pebbling formula has critical block sensitivity $\geq d/2$

How do we determine the edges of G?



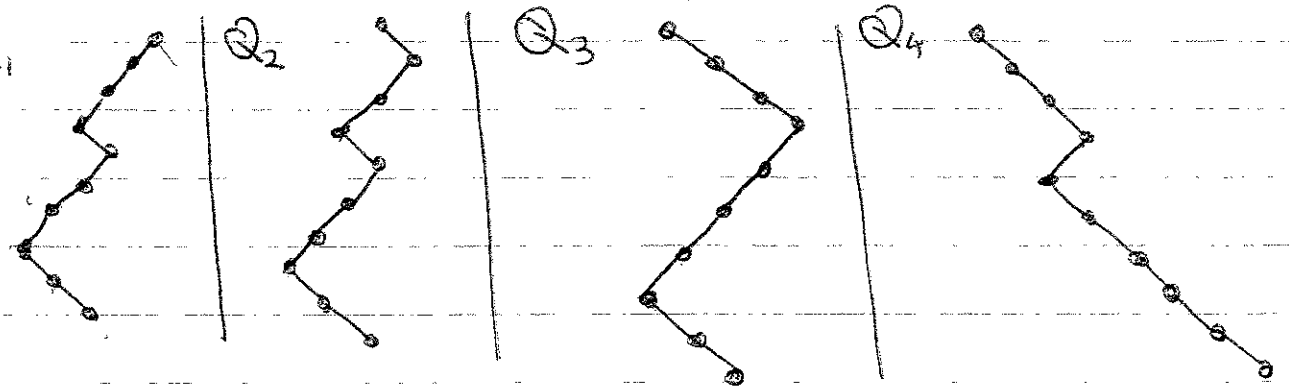
given a path P we consider the lowest level where the path has horizontal coordinate of a specific value

In this example there are 6 nodes in the path such that the path does not reach the same position anymore

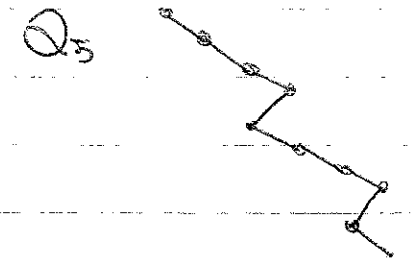
For each such position in path P we define path $Q_1, Q_2, \ldots$ such that

- P and $Q_i$ are identical from sink to that node,
- then P and $Q_i$ do opposite moves

E.g. $Q_1$ $Q_2$ $Q_3$ $Q_4$



16




We add $\{P, Q_i\}$ to $E(G)$.

Obs Notice that this is well defined, since

P produce Q iff Q produce P in this construction.

Notice that each path P induced a critical assignment, so any edge in $E(G)$ induces a

BICRITICAL ASSIGNMENT $(P, Q)$

and the search problem on the assignment $(P, Q)$ answers either SOURCE$(P)$ or SOURCE$(Q)$.

the graph G has the following characteristic

- if $\{P, Q_1\}$ and $\{P, Q_2\}$ are edges then

$$Q_1 \cap Q_2 \subseteq P \quad [\text{the are parallel}]$$

  - the assignment corresponding to P can be transformed in one corresponding to $(P, Q)$ by flipping some bits.

- if $\alpha$ is the assignment corresponding to P and $\alpha_i$ corresponds to $(P, Q_i)$ $\forall Q_i \in \Gamma(P)$ then each $\alpha_i = \alpha^{B_i}$ for some $B_i \in Vars(P_n)$

- $\boxed{B_i \cap B_j = \emptyset \quad \forall\, Q_i, Q_j \in \Gamma(P)}$

- If the answer to the search problem would change from $\alpha$ to each $\alpha_i$

we <u>would have</u> $bs_{orit}(Search(P_n)) \geq deg(G)$

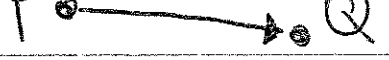
But the search problem is solved by some function which wants to reduce block sensitivity.

Given any edge $\{P, Q\}$ the function either answer

Search(P) or Search(Q).

We ORIENT THE GRAPH according to f

**Intuition**




Moving from assignment $\alpha_Q$ to $\alpha_{(P,Q)}$ does not change the value of the output, but moving from $\alpha_P$ to $\alpha_{(P,Q)}$ does

So the edge contributes to the block sensitivity of at least one assignment.

**Lemma 2** If path graph G has average degree at least $d$

then $bs_{crit}(Search(P_n)) \geq \frac{d}{2}$

**Proof** Let be $d_p$, $d_p^-$, $d_p^+$ the degree, indegree, out degree of the vertex corresponding to path P

$$d \cdot |V(G)| = \sum_P d_p = \sum_P (d_p^+ + d_p^-) = \left(\sum_P d_p^+\right) + \left(\sum_P d_p^-\right)$$

But $\sum_P (d_p^+ - d_p^-) = \emptyset$

$$\Rightarrow \sum_P d_p^+ = \frac{d}{2} \cdot |V(G)|$$

So there is at least a verter P with out degree $\frac{d}{2}$, so the corresponding assignment $\alpha_p$ has

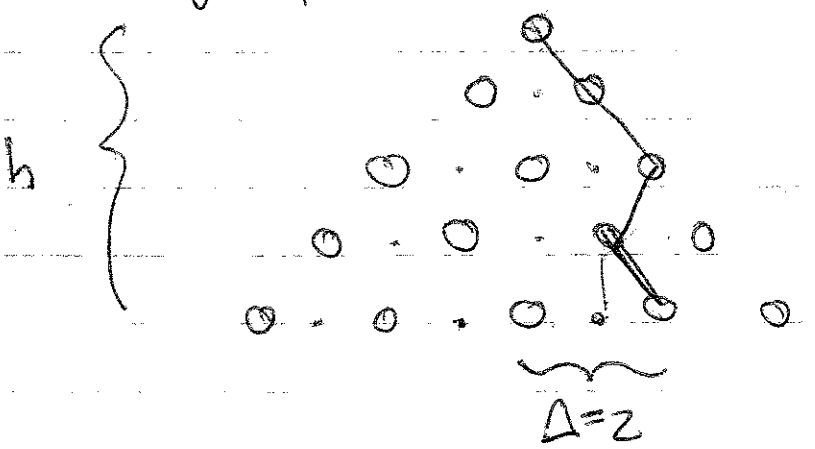
$$bs(f, \alpha_p) \geq \frac{d}{2}$$

$$\boxed{bs_{crit}(Search(P_n)) \geq \frac{d}{2}}$$

□

to show the corollary 4 we need to show that the path graph has average degree $4\sqrt{n}$.

Obs If a path P ends in a source node at distance $\Delta$ from the center then it has AT LEAST $\Delta$ neighbors in the path graph




Paths as random walks:

they start at the center and go left or right with uniform probability!

~~They can be seen~~

$$\Delta = \sum_{i=1}^{h} X_i \quad \text{where} \quad X_i = \begin{cases} -1 & \text{w.p. } 1/2 \\ +1 & \text{w.p. } 1/2 \end{cases}$$

- Each outcome of $X_1 \cdots X_h$ corresponds to a path.
- $|\Delta|$ corresponds to the distance of the source node of this path to the center

**Fact** [Erdős, 45]

there are $\beta, \gamma > 0$ s.t.

$$P_r\left[|\Delta| > \gamma\sqrt{h}\right] > \beta$$

So

$$|V(G)| \cdot \frac{d}{2} = \sum_P d_P^+ \geq \beta |V(G)| \cdot \gamma\sqrt{h}$$

↳ $\exists$ a vertex with outdegree

$$\geq \Omega(\sqrt{h}) = \Omega(\sqrt[4]{n})$$

□

# Comments on the tradeoff

We know that in Resolution,

$$\text{Space}(F_n) = \Omega(\sqrt{n}) .$$

so knowing that

$$\text{Space}(F_n) \cdot \log \text{Length}(F_n) \gtrsim \Omega(\sqrt[4]{n})$$

does not give us tradeoffs

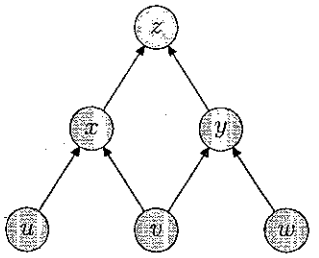
We don't know if the space in CP and PCR is large for formulas $F_n$ based on pyramid graphs, but we suspect so.

What we would like is $F_n$ such that

- $F_n$ refutable in short time
- $F_n$ refutable in small space

- If $F_n$ is refuted in space $S$ and length $L$

$$S \log L \not\geq \text{poly}(n)$$




(a) Pyramid graph $\Pi_2$ of height 2.

$$
\begin{aligned}
& u \\
\wedge\ & v \\
\wedge\ & w \\
\wedge\ & (\overline{u} \vee \overline{v} \vee x) \\
\wedge\ & (\overline{v} \vee \overline{w} \vee y) \\
\wedge\ & (\overline{x} \vee \overline{y} \vee z) \\
\wedge\ & \overline{z}
\end{aligned}
$$

(b) Pebbling contradiction $Peb_{\Pi_2}$.

**Figure 1:** Pebbling contradiction for the pyramid graph $\Pi_2$.

$$
\begin{aligned}
& (y_{u,1} \vee y_{u,2}) \\
\wedge\ & (y_{v,1} \vee y_{v,2}) \\
\wedge\ & (y_{w,1} \vee y_{w,2}) \\
\wedge\ & (y_{x,1} \vee y_{x,2}) \\
\wedge\ & (y_{y,1} \vee y_{y,2}) \\
\wedge\ & (y_{z,1} \vee y_{z,2}) \\
\wedge\ & (\overline{y}_{u,1} \vee x_{u,1}) \\
\wedge\ & (\overline{y}_{u,2} \vee x_{u,2}) \\
\wedge\ & (\overline{y}_{v,1} \vee x_{v,1}) \\
\wedge\ & (\overline{y}_{v,2} \vee x_{v,2}) \\
\wedge\ & (\overline{y}_{w,1} \vee x_{w,1}) \\
\wedge\ & (\overline{y}_{w,2} \vee x_{w,2}) \\
\wedge\ & (\overline{y}_{u,1} \vee \overline{x}_{u,1} \vee \overline{y}_{v,1} \vee \overline{x}_{v,1} \vee \overline{y}_{x,1} \vee x_{x,1}) \\
\wedge\ & (\overline{y}_{u,1} \vee \overline{x}_{u,1} \vee \overline{y}_{v,1} \vee \overline{x}_{v,1} \vee \overline{y}_{x,2} \vee x_{x,2}) \\
\wedge\ & (\overline{y}_{u,1} \vee \overline{x}_{u,1} \vee \overline{y}_{v,2} \vee \overline{x}_{v,2} \vee \overline{y}_{x,1} \vee x_{x,1}) \\
\wedge\ & (\overline{y}_{u,1} \vee \overline{x}_{u,1} \vee \overline{y}_{v,2} \vee \overline{x}_{v,2} \vee \overline{y}_{x,2} \vee x_{x,2}) \\
\wedge\ & (\overline{y}_{u,2} \vee \overline{x}_{u,2} \vee \overline{y}_{v,1} \vee \overline{x}_{v,1} \vee \overline{y}_{x,1} \vee x_{x,1}) \\
\wedge\ & (\overline{y}_{u,2} \vee \overline{x}_{u,2} \vee \overline{y}_{v,1} \vee \overline{x}_{v,1} \vee \overline{y}_{x,2} \vee x_{x,2}) \\
\wedge\ & (\overline{y}_{u,2} \vee \overline{x}_{u,2} \vee \overline{y}_{v,2} \vee \overline{x}_{v,2} \vee \overline{y}_{x,1} \vee x_{x,1}) \\
\wedge\ & (\overline{y}_{u,2} \vee \overline{x}_{u,2} \vee \overline{y}_{v,2} \vee \overline{x}_{v,2} \vee \overline{y}_{x,2} \vee x_{x,2})
\end{aligned}
$$

$$
\begin{aligned}
\wedge\ & (\overline{y}_{v,1} \vee \overline{x}_{v,1} \vee \overline{y}_{w,1} \vee \overline{x}_{w,1} \vee \overline{y}_{y,1} \vee x_{y,1}) \\
\wedge\ & (\overline{y}_{v,1} \vee \overline{x}_{v,1} \vee \overline{y}_{w,1} \vee \overline{x}_{w,1} \vee \overline{y}_{y,2} \vee x_{y,2}) \\
\wedge\ & (\overline{y}_{v,1} \vee \overline{x}_{v,1} \vee \overline{y}_{w,2} \vee \overline{x}_{w,2} \vee \overline{y}_{y,1} \vee x_{y,1}) \\
\wedge\ & (\overline{y}_{v,1} \vee \overline{x}_{v,1} \vee \overline{y}_{w,2} \vee \overline{x}_{w,2} \vee \overline{y}_{y,2} \vee x_{y,2}) \\
\wedge\ & (\overline{y}_{v,2} \vee \overline{x}_{v,2} \vee \overline{y}_{w,1} \vee \overline{x}_{w,1} \vee \overline{y}_{y,1} \vee x_{y,1}) \\
\wedge\ & (\overline{y}_{v,2} \vee \overline{x}_{v,2} \vee \overline{y}_{w,1} \vee \overline{x}_{w,1} \vee \overline{y}_{y,2} \vee x_{y,2}) \\
\wedge\ & (\overline{y}_{v,2} \vee \overline{x}_{v,2} \vee \overline{y}_{w,2} \vee \overline{x}_{w,2} \vee \overline{y}_{y,1} \vee x_{y,1}) \\
\wedge\ & (\overline{y}_{v,2} \vee \overline{x}_{v,2} \vee \overline{y}_{w,2} \vee \overline{x}_{w,2} \vee \overline{y}_{y,2} \vee x_{y,2}) \\
\wedge\ & (\overline{y}_{x,1} \vee \overline{x}_{x,1} \vee \overline{y}_{y,1} \vee \overline{x}_{y,1} \vee \overline{y}_{z,1} \vee x_{z,1}) \\
\wedge\ & (\overline{y}_{x,1} \vee \overline{x}_{x,1} \vee \overline{y}_{y,1} \vee \overline{x}_{y,1} \vee \overline{y}_{z,2} \vee x_{z,2}) \\
\wedge\ & (\overline{y}_{x,1} \vee \overline{x}_{x,1} \vee \overline{y}_{y,2} \vee \overline{x}_{y,2} \vee \overline{y}_{z,1} \vee x_{z,1}) \\
\wedge\ & (\overline{y}_{x,1} \vee \overline{x}_{x,1} \vee \overline{y}_{y,2} \vee \overline{x}_{y,2} \vee \overline{y}_{z,2} \vee x_{z,2}) \\
\wedge\ & (\overline{y}_{x,2} \vee \overline{x}_{x,2} \vee \overline{y}_{y,1} \vee \overline{x}_{y,1} \vee \overline{y}_{z,1} \vee x_{z,1}) \\
\wedge\ & (\overline{y}_{x,2} \vee \overline{x}_{x,2} \vee \overline{y}_{y,1} \vee \overline{x}_{y,1} \vee \overline{y}_{z,2} \vee x_{z,2}) \\
\wedge\ & (\overline{y}_{x,2} \vee \overline{x}_{x,2} \vee \overline{y}_{y,2} \vee \overline{x}_{y,2} \vee \overline{y}_{z,1} \vee x_{z,1}) \\
\wedge\ & (\overline{y}_{x,2} \vee \overline{x}_{x,2} \vee \overline{y}_{y,2} \vee \overline{x}_{y,2} \vee \overline{y}_{z,2} \vee x_{z,2}) \\
\wedge\ & (\overline{y}_{z,1} \vee \overline{x}_{z,1}) \\
\wedge\ & (\overline{y}_{z,2} \vee \overline{x}_{z,2})
\end{aligned}
$$

**Figure 2:** Lifted formula $Lift_2(Peb_{\Pi_2})$ of length 2 obtained from the pebbling contradiction over $\Pi_2$.