# The PPSZ algorithm

**Seminar on exact exponential algorithms**

Tal Lancewicki

Based on Paturi et al. [1998]

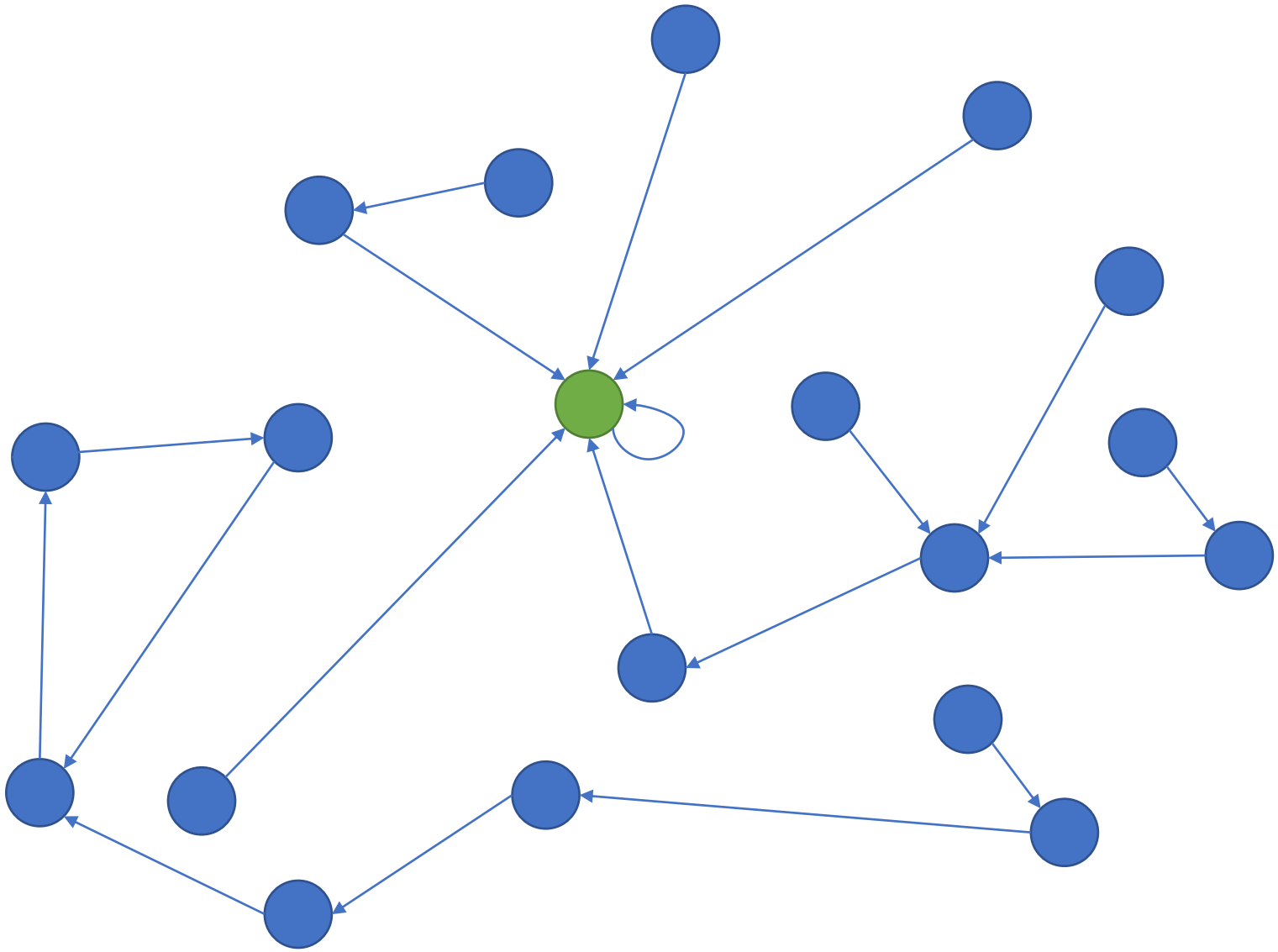
# K-SAT problem

- CNF with $n$ variables
- Each clause has at most $k$ literals
- Is there a satisfying solution?

$$G = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_3} \vee \overline{x_6} \vee x_5)$$

# Overview

| $k$ | PPSZ Unique-k-SAT | PPSZ General k-SAT | Schöning [1999] | Hertli [2011] | Hansen et al. [2019] |
|---|---|---|---|---|---|
| 3 | $1.30704^n$ | $1.434^n$ | $1.334^n$ | $1.30704^n$ | $1.306995^n$ |
| 4 | $1.468^n$ | $1.477^n$ | $1.5^n$ | $1.469^n$ | |
| 5 | $1.569^n$ | | $1.6^n$ | - | |
| 6 | $1.637^n$ | | $1.667^n$ | - | |



# Restriction of a formula

- $G = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$

$$G|_{x_1=1} = \cancel{(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge} (x_2 \vee \overline{x_3} \vee \overline{x_4}) \wedge (\cancel{\overline{x_1} \vee} \overline{x_2} \vee x_3)$$

$$= (x_2 \vee \overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_2} \vee x_3)$$

$$G|_{x_1=0} = (\cancel{x_1 \vee} \overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4}) \wedge \cancel{(\overline{x_1} \vee \overline{x_2} \vee x_3)}$$

$$= (\overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4})$$

Function $Modify(G, \pi, y)$

- for $i \in [n]$:
  - If $x_{\pi(i)}$ or $\bar{x}_{\pi(i)}$ appears in $G$ as unit clause:
    - Set $u_{\pi(i)}$ to 1 or 0, respectively.
  - Else
    - Set $u_{\pi(i)} = y_{\pi(i)}$
  - $G := G|_{x_{\pi(i)}=u_{\pi(i)}}$
- return $u$

Function $Search(G, I)$

- for $I$ times:
  - $\pi \sim U(S_n)$
  - $y \sim U(\{0,1\}^n)$
  - $u = Modify(G, \pi, y)$
  - If $u$ satisfy $G$, return $u$
- return "unsatisfiable"

# *Modify* function example

$\pi = id, y = (1,1,0,0)$

$G =$

$(x_1 \vee \overline{x_2} \vee \overline{x_3})$
$\wedge (x_1 \vee x_2 \vee \overline{x_4})$
$\wedge (x_1 \vee \overline{x_2} \vee \overline{x_4})$
$\wedge (x_2 \vee \overline{x_3} \vee \overline{x_4})$
$\wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$
$\wedge (\overline{x_1} \vee \overline{x_2} \vee x_4)$
$\wedge (\overline{x_1} \vee x_2 \vee x_4)$
$\wedge (\overline{x_1} \vee x_2 \vee \overline{x_4})$
$\wedge (x_1 \vee x_2 \vee x_4)$
$\wedge (x_1 \vee x_3 \vee x_4)$

$\rightarrow$

$u_1 = 1$

$G =$

$(x_2 \vee \overline{x_3} \vee \overline{x_4})$
$\wedge (\overline{x_2} \vee x_3)$
$\wedge (\overline{x_2} \vee x_4)$
$\wedge (x_2 \vee x_4)$
$\wedge (x_2 \vee \overline{x_4})$

$\rightarrow$

$u_2 = 1$

$G =$

$(x_3)$
$\wedge (x_4)$

$\rightarrow$

$u_3 = 1$

$G =$

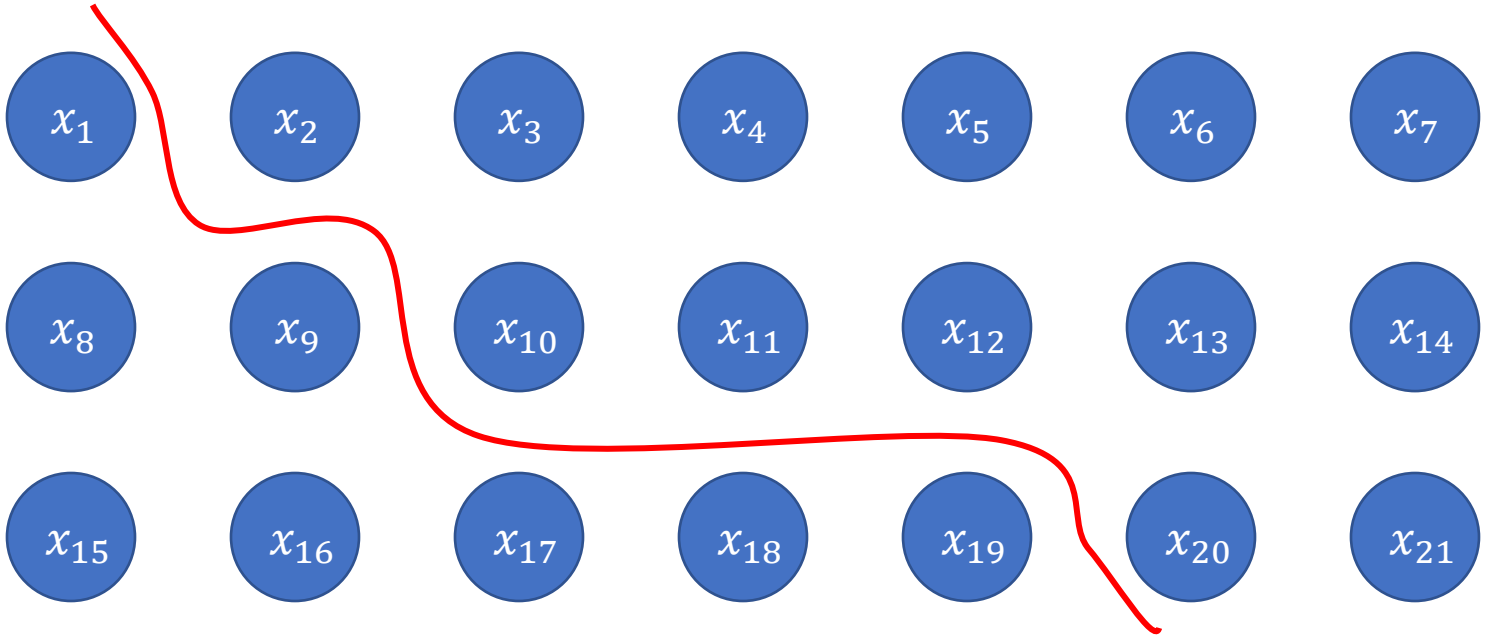
$(x_4)$

$\rightarrow$

$u_4 = 1$

$G =$

# $Modify$ splits the variable space




- Denote by $Forced(\pi, y)$ the set of variables that are forced by $Modify(\pi, y)$.

# Carchtarizing $Modify^{-1}(z)$

**<u>Lemma 1</u>**: If $z$ is a satisfying assignment then $Modify(\pi, y) = z$ if and only if $y$ and $z$ agree on all variables outside $Forced(\pi, z)$.

$z = (1,0,1,0,1,1)$

$Forced = \{x_1, x_3, x_4\}$

| Variable | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| Possible $y$ | 0,1 | 0 | 0,1 | 0,1 | 1 | 1 |

# Proof

- W.l.o.g assume that $z = 1^n$ and $\pi = id$.
- Assume $y$ agrees with $z$ on all variables outside of forced.
  - Let $i$ be the first index that $y$ is 0. Then $x_i \in Forced(\pi, z)$. Also, until step $i$ $Modify$ assigned the same variables given $y$ and given $z$. Since $z$ is satisfiable, $x_i$ must be forced to 1.
  - Continue by induction on the zeros of $y$.
- Assume $y_i \neq z_i$ $(y_i = 0)$ and that $x_i \notin Forced(\pi, z)$ for some $i$.
  - Let $i$ be the first that satisfies the above. Then $x_i$ is not forced under $y$ as well.
  - Hence, we'll set it to $0$.

- Let $z$ be a **unique** satisfying assignment.
- Let $\tau$ the probability with respect to random $\pi$ and $y$ that $Modify(\pi, y) = z$.

**<u>Lemma 2</u>**: We can bound $\tau$ by

$$\tau \geq 2^{-n+E_{\pi}[|Forced(\pi,z)|]}$$

# Proof of lemma 2

By lemma 1, for fixed $\pi$

$$P(Modify(G,\pi,y)=z)=\frac{2^{|Forced(\pi,z)|}}{2^n}$$

Taking the expectation over $\pi$

$$\tau=2^{-n}E_\pi\left[2^{|Forced(\pi,z)|}\right]\geq 2^{-n+E_\pi[|Forced(\pi,z)|]}$$

Where the last inequality if Jensen's inequality for the convex function $2^x$

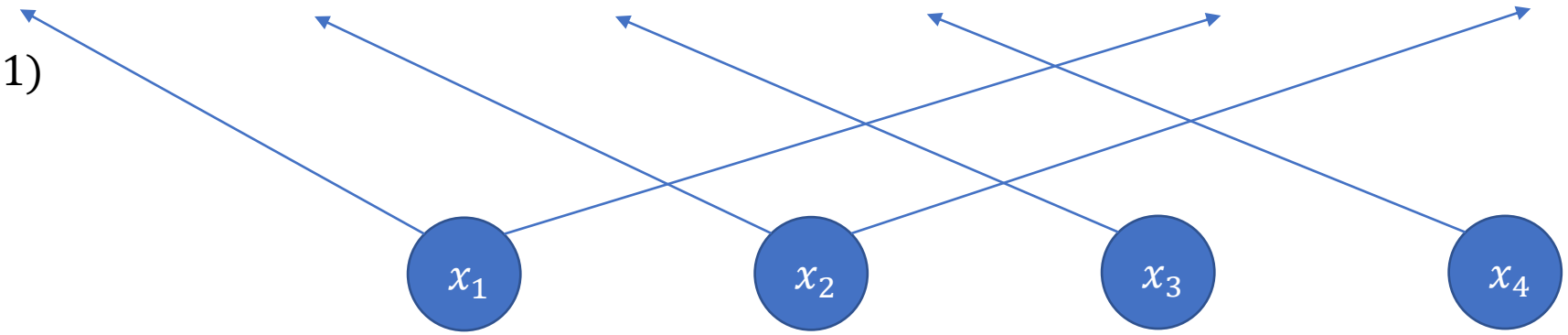
# Critical clauses

- We say that $C$ is a critical clause for $v$ if
  - $v \in C$, and
  - Flipping the value of $v$ at $z$, flips $C$ as well.

➢ If $z$ is unique then each variable have at least one critical clause.

$$G = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_2 \vee x_4)$$

$z = (1,1,1,1)$



- Note that $v \in Forced(G, \pi, z)$ if and only if $v$ is the last variable with respect to $\pi$ in some critical clause.

$$(x_1 \vee \overline{x_2} \vee \overline{x_3}) \qquad (x_1 \vee x_2 \vee \overline{x_3})$$

- What can we say about $P_\pi\ (v \in Forced(G, \pi, z))$ when $z$ is unique?

$$P_v := P_\pi\left(v \in Forced(\pi, z)\right) \geq \frac{1}{k}$$

Hence,

$$\begin{aligned}
\tau &\geq 2^{-n+E_\pi[|Forced(\pi,z)|]} \\
&= 2^{-n+\sum_v E_\pi\left[1_{\{v \in Forced(\pi,z)\}}\right]} \\
&= 2^{-n+\sum_v P_v} \\
&\geq 2^{-\left(1-\frac{1}{k}\right)n}
\end{aligned}$$

# PPZ Algorithm for unique SAT

$\mathrm{PPZ}(F)$

- For $n2^{\left(1-\frac{1}{k}\right)n}$ times:
  - $\pi \sim U(S_n)$
  - $y \sim U(\{0,1\}^n)$
  - $u = Modify(\pi, y)$
  - If $u$ satisfy $F$, return $u$
- return "unsatisfiable"

- If there is no solution then PPZ return "unsatisfiable"
- If there is a <u>unique solution</u> we return "unsatisfiable" with probability at most

$$\left(1 - 2^{-\left(1-\frac{1}{k}\right)n}\right)^{n2^{\left(1-\frac{1}{k}\right)n}} \leq e^{-n}$$

❖ $(1 - x \leq e^{-x})$

- For k=3 run time is $O^*(1.587^n)$

We only used the fact that there exist a critical clause for each variable

# Adding critical clauses

- One critical clause

$$C_1 = (x_1 \vee \overline{x_2} \vee \overline{x_3})$$

  - $P_1 = {}^1/_3$
- Two critical clauses

$$C_2 = (x_1 \vee \overline{x_4} \vee \overline{x_5})$$

  - $P_1 = {}^7/_{15}$
- $F$ might contain <u>only</u> 1 critical clause per variable

- Consider $C_1 = (x_1 \vee \overline{x_2} \vee \overline{x_3})$ and $C_2 = (x_2 \vee \overline{x_4} \vee \overline{x_5})$
- Lets add to $F$ the clause $(x_1 \vee \overline{x_3} \vee \overline{x_4} \vee \overline{x_5})$
- Another critical clause without changing the set of solutions!
- Can we always add critical clauses?

# Resolving clauses

- We say that $C_1$ and $C_2$ are conflict on $v$, if one contains $v$ and the other contain $\bar{v}$.

$$(x_1 \vee \overline{x_2} \vee \overline{x_3}), (x_2 \vee \overline{x_4} \vee \overline{x_5})$$

- We say that $C_1$ and $C_2$ are resolvable if they conflict on exactly one variable.
- In the case above, the resolvent of $C_1$ and $C_2$ is defined by

$$R(C_1, C_2) := D_1 \vee D_2$$

where $D_i$ is obtained by omitting the variable $v$ from $C_i$

$$(x_1 \vee \overline{x_3} \vee \overline{x_4} \vee \overline{x_5})$$

# Can we always add critical clauses?

- Assume $z = 1^n$ is a unique solution
- $C_1 = (x_1 \vee \overline{x_2} \vee \overline{x_3})$

- So consider a clause that is not satisfied under $001^{n-2}$

$\boldsymbol{x_2 \in C_2}$**:**

$C_2 = (x_2 \vee \overline{x_4} \vee \overline{x_5})$

➢$R(C_1, C_2) = (x_1 \vee \overline{x_3} \vee \overline{x_4} \vee \overline{x_5})$

$C_2 = (x_1 \vee x_2 \vee \overline{x_5})$

➢$R(C_1, C_2) = (x_1 \vee \overline{x_3} \vee \overline{x_5})$

$\boldsymbol{x_2 \notin C_2}$**:**

$C_2 = (x_1 \vee \overline{x_4} \vee \overline{x_5})$.

That is another critical clause for $x_1$.

# PPSZ Algorithm

Function $Resolve(F, s)$

- while $F$ has a resolvable pair such that $R(C_1, C_2) \notin F$ and $|R(C_1, C_2)| \leq s$:
  - $F := F \wedge R(C_1, C_2)$
- Return $F$

- This adds at most $(2n)^s$ new clauses
- If $s = o({}^{n}/_{\log(n)})$ then $n^s = 2^{o(n)}$

$\mathrm{PPSZ}(F)$

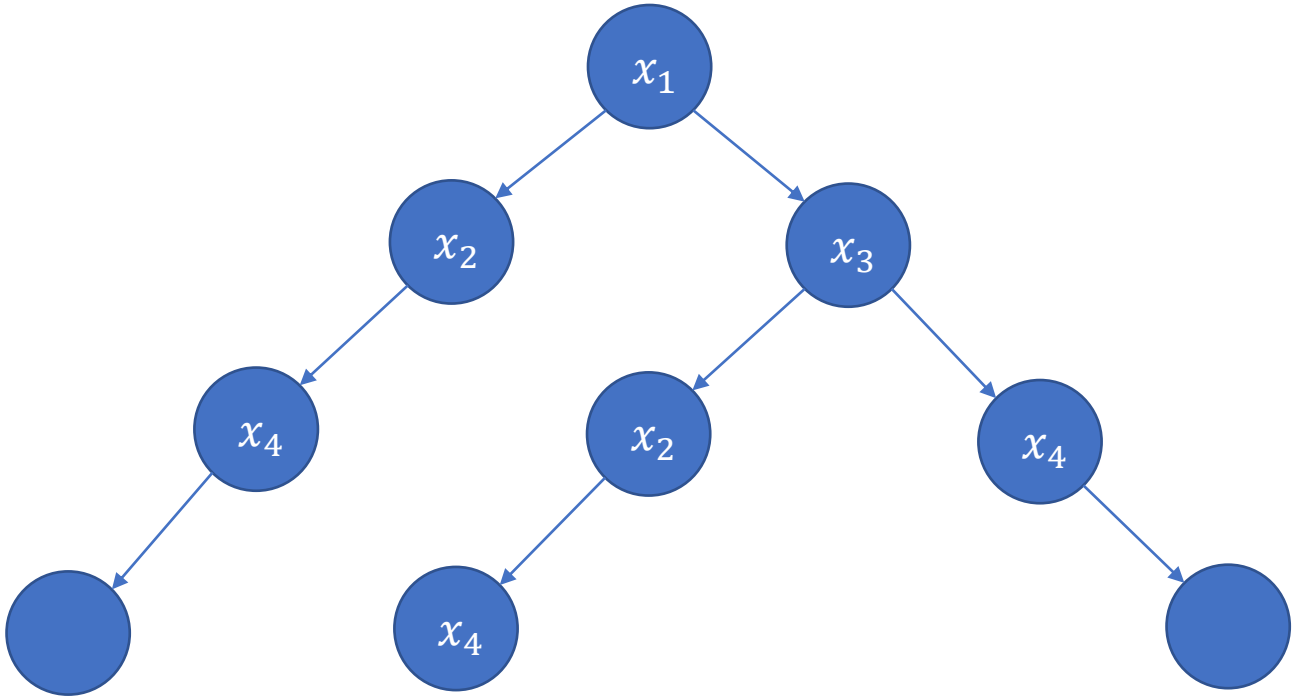
- $F_s = \mathrm{Resolve}(F, s)$
- For $I$ times:
  - $\pi \sim U(S_n)$
  - $y \sim U(\{0,1\}^n)$
  - $u = Modify(F_s, \pi, y)$
  - If $u$ satisfy $F$, return $u$
- return "unsatisfiable"

- $s$ and $I$ will be determined later

We need a way to track how much critical clauses Resolve add

# Admissible tree

- We say that a rooted tree is admissible if
  - The root is labeled as variable
  - The rest of the nodes are either labeled as variable or unlabeled
  - For any path from the root to a leaf, each variable appears at most once



# Critical clause tree

- A cut is a set of nodes $A$ that <u>does not include the root</u>, and any path from the root to a leaf goes throw $A$.

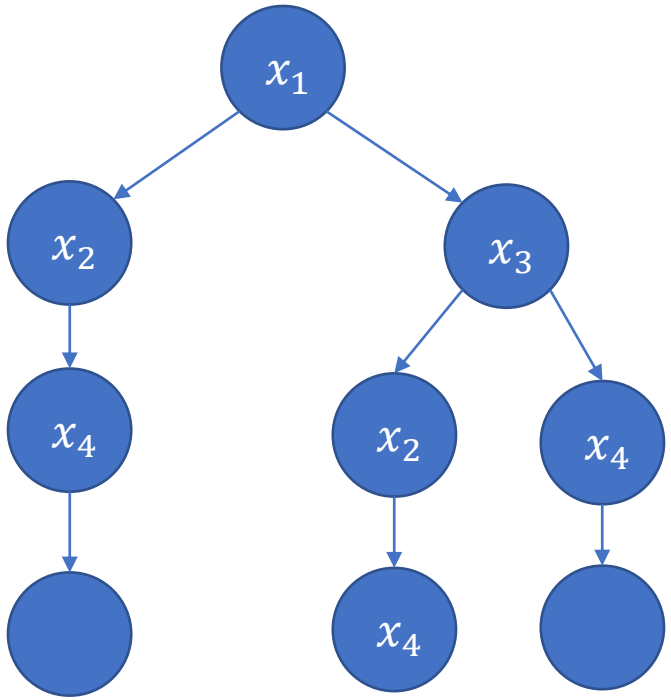
- We say that an admissible tree is a critical clause tree for $(v, G, z)$ if
  - The root is labeled with $v$
  - For any cut $A$, there exist a critical clause $C(A)$ for $(v, G, z)$ such that

$$var\big(C(A)\big) \subseteq var(A) \cup \{v\}$$

# Example

- We say that an admissible tree is a critical clause tree for $v$ if
  - The root is labeled with $v$
  - For any cut $A$, there exist a critical clause $C(A)$ for $v$ such that

$$var\big(C(A)\big) \subseteq var(A) \cup \{v\}$$




$F =$

$\boxed{(x_1 \vee \overline{x_2} \vee \overline{x_3})}$

$(x_1 \vee x_2 \vee \overline{x_4})$

$\boxed{(x_1 \vee \overline{x_2} \vee \overline{x_4})}$

$(x_2 \vee \overline{x_3} \vee \overline{x_4})$

$(\overline{x_1} \vee \overline{x_2} \vee x_3)$

$(\overline{x_1} \vee \overline{x_2} \vee x_4)$

$(\overline{x_1} \vee x_2 \vee x_4)$

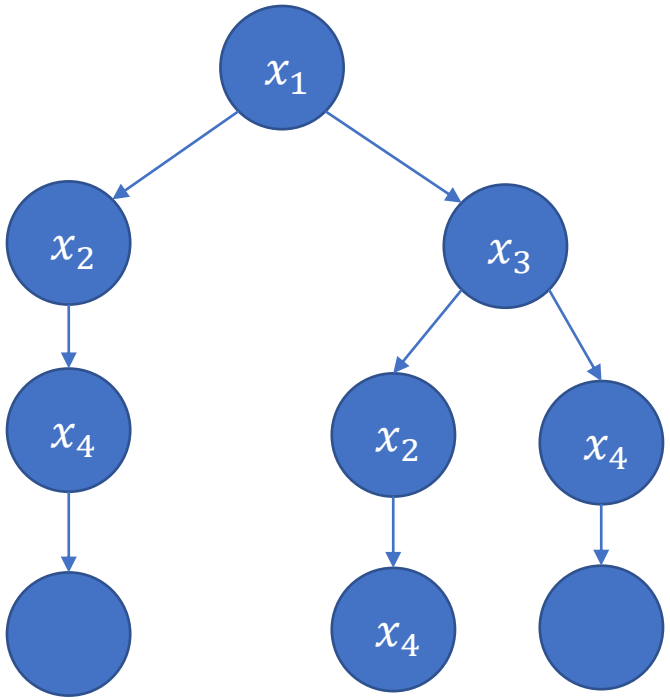
$(\overline{x_1} \vee x_2 \vee \overline{x_4})$

$(x_1 \vee x_2 \vee x_4)$

$(x_1 \vee x_3 \vee x_4)$

Not a critical clause tree with respect to $F$!

# What about $F_s$?




$F =$

$(x_1 \vee \overline{x_2} \vee \overline{x_3})$

$(x_1 \vee x_2 \vee \overline{x_4})$

$(x_1 \vee \overline{x_2} \vee \overline{x_4})$

$(x_2 \vee \overline{x_3} \vee \overline{x_4})$

$(\overline{x_1} \vee \overline{x_2} \vee x_3)$

$(\overline{x_1} \vee \overline{x_2} \vee x_4)$

$(\overline{x_1} \vee x_2 \vee x_4)$

$(\overline{x_1} \vee x_2 \vee \overline{x_4})$

$(x_1 \vee x_2 \vee x_4)$

$(x_1 \vee x_3 \vee x_4)$

$(x_1 \vee \overline{x_4})$

$(x_1 \vee \overline{x_3} \vee x_4)$

$(\overline{x_1} \vee x_4)$

$(x_1 \vee \overline{x_3})$

$(\overline{x_3} \vee x_4)$

$(x_1 \vee x_4)$

$(x_1)$

A critical clause tree for $F_s$

# Existence of critical clause tree

**<u>Lemma 4</u>**

For any variable $v$ and $s \geq k^d$ ($d \leq n$) there exist a critical clause tree for $(v, F_s)$ of maximum degree $k-1$ and uniform depth $d$.

- $d$ will be determined later

# Proof of Lemma 4

Fix $v$ and $z$. W.l.o.g $z = 1^n$. We'll grow a tree in following way:

- $T_0$ consist of the node $v$
- Given $T_{i-1}$, if all the leaves are of depth $d$, stop. Otherwise,
  - Choose a leaf $b_i$ of depth < $d$
  - Let $P_i$ the path from the root to $b_i$.
  - $z \oplus P_i$ is not a solution!
  - Let $C_i$ be a clause that is not satisfied under $z \oplus P_i$
  - Add the variables $var(C_i) \backslash \mathrm{P}_i$ as childes of $b_i$
    - If empty add unlabeled child

# Proof cont

- Choose a leaf $b_i$ of depth < $d$
- Let $P_i$ the path from the root to $b_i$.
- $C_i$ is not satisfied under $z \oplus P_i$
- Add $var(C_i) \backslash \mathrm{P}_i$ as childes of $b_i$
  - If empty add unlabeled child

$$\boldsymbol{z} = (\mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1})$$

$$\boldsymbol{z} \oplus \boldsymbol{P_i} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$$

$F =$

$(x_1 \vee \overline{x_2} \vee \overline{x_3})$

$(x_1 \vee x_2 \vee \overline{x_4})$

$(x_1 \vee \overline{x_2} \vee \overline{x_4})$

$(x_2 \vee \overline{x_3} \vee \overline{x_4})$

$(\overline{x_1} \vee \overline{x_2} \vee x_3)$

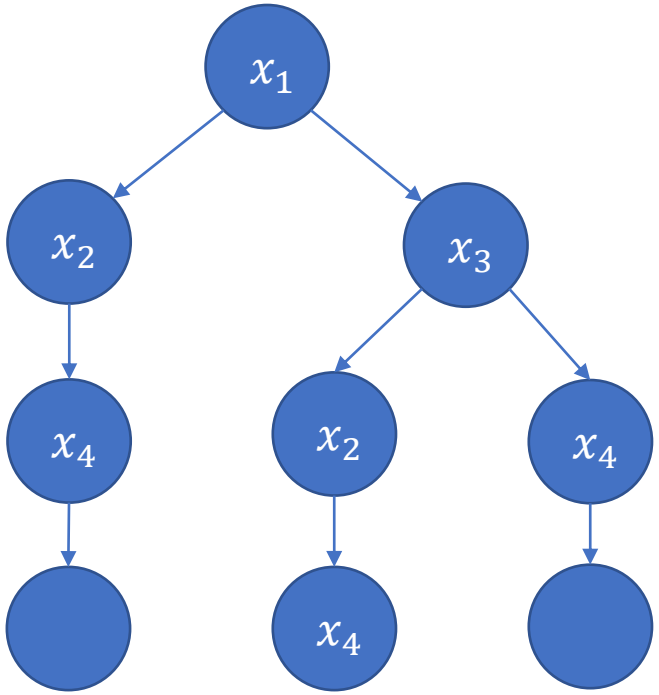
$(\overline{x_1} \vee \overline{x_2} \vee x_4)$

$(\overline{x_1} \vee x_2 \vee x_4)$

$(\overline{x_1} \vee x_2 \vee \overline{x_4})$

$(x_1 \vee x_2 \vee x_4)$

$(x_1 \vee x_3 \vee x_4)$



# Proof cont

- Choose a leaf $b_i$ of depth < $d$
- Let $P_i$ the path from the root to $b_i$.
- $C_i$ is not satisfied under $z \oplus P_i$
- Add $var(C_i) \backslash \mathrm{P}_i$ as childes of $b_i$
  - If empty add unlabeled child

- We'll show that $T_i$ is critical clause tree for any $i$
- A tree is admissible if
  - ✓ The root is labeled as variable
  - ✓ The rest of the nodes are either labeled as variable or unlabeled
  - ✓ For any path from the root to a leaf, each variable appears at most one time
- At the end of the process we get also uniform depth $d$
- Remains to show that for any cut $A$, there exist a critical clause $C(A)$ for $(v, F_s)$ such that

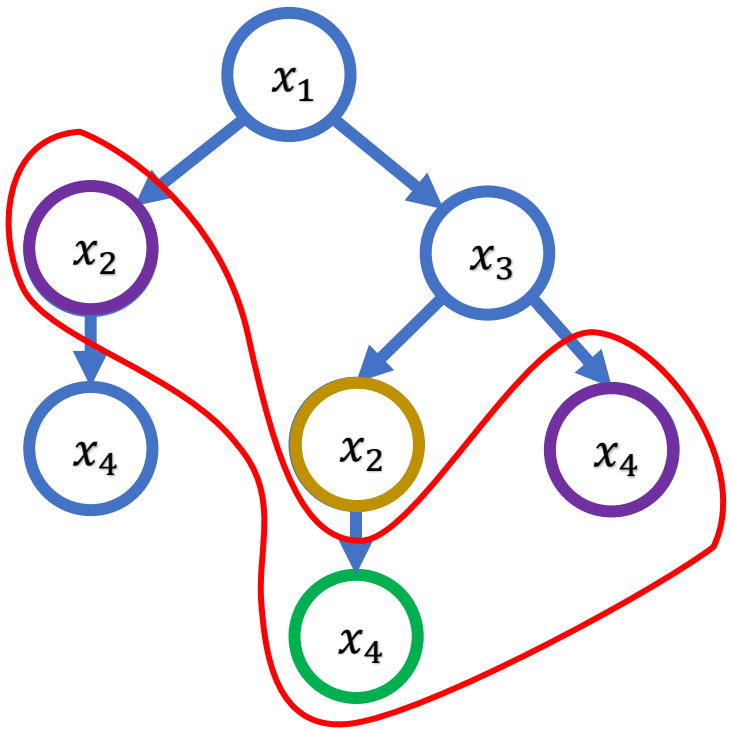
$$var\big(C(A)\big) \subseteq var(A) \cup \{v\}$$

# Proof cont

- $T_0$ and $T_1$ is trivial
- Let $\boldsymbol{N = T_i \backslash T_{i-1}}$
- Let $\boldsymbol{A}$ a cut in $T_i$ (w.l.o.g $A$ is minimal)
- Let $\boldsymbol{A' = A \backslash N}$ (w.l.o.g $N \subseteq A$)
- For any $j \in P_i \backslash \{v\}$, $A_j = A' \cup \{j\}$ is a cut in $\boldsymbol{T_{i-1}}$
- **<u>Case 1</u>**: If for some $\boldsymbol{j, var(A_j) \subseteq var(A')}$ then

$$\exists C(A_j) \in F_s:$$

$$\begin{aligned} var\left(C(A_j)\right) &\subseteq var(A_j) \cup \{v\} \\ &\subseteq var(A') \cup \{v\} \\ &\subseteq var(A) \cup \{v\} \end{aligned}$$




$F =$

$(x_1 \vee \overline{x_2} \vee \overline{x_3})$

$(x_1 \vee x_2 \vee \overline{x_4})$

$(x_1 \vee \overline{x_2} \vee \overline{x_4})$

$(x_2 \vee \overline{x_3} \vee \overline{x_4})$

$(\overline{x_1} \vee \overline{x_2} \vee x_3)$

$(\overline{x_1} \vee \overline{x_2} \vee x_4)$

$(\overline{x_1} \vee x_2 \vee x_4)$

$(\overline{x_1} \vee x_2 \vee \overline{x_4})$

$(x_1 \vee x_2 \vee x_4)$

$(x_1 \vee x_3 \vee x_4)$

$A_j = A' \cup \{j\}$

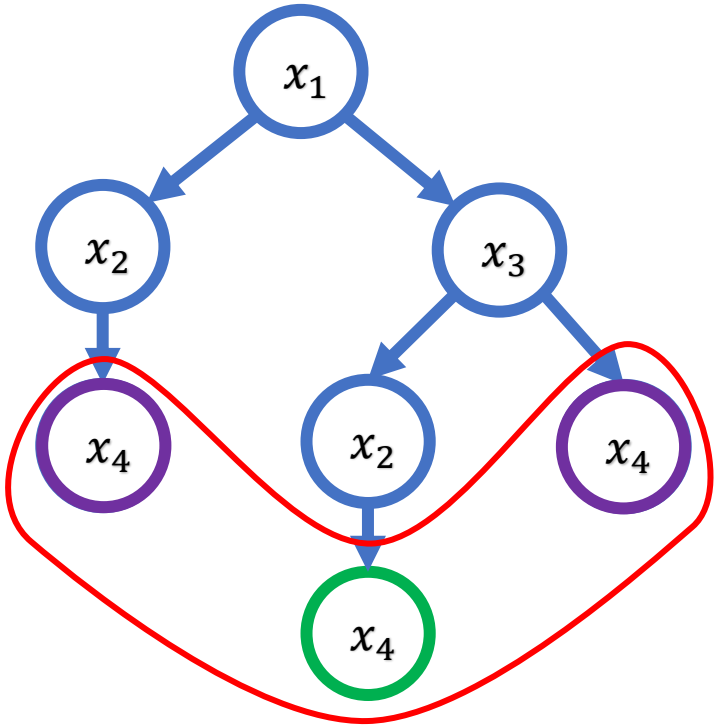
$\boldsymbol{N = T_i \backslash T_{i-1}}$

$\boldsymbol{A}$ a cut in $T_i$

$\boldsymbol{A' = A \backslash N}$

# Proof cont

- **<u>Case 2</u>**: For all $j \in P_i$, $var(A_j)$ $\not\subseteq$ $\boldsymbol{var(A')}$
  i.e. $r_j \notin \mathrm{var}(\mathrm{A}')$ (where $r_j$ is $j$'s variable)
- Let the clause $\boldsymbol{C_i}$ that was used in the construction of $T_i$
- $C_i = R \vee U$ where $R$ consist <u>negation</u> of variables from $N$, and $U \subseteq P_i$ <u>(positively)</u>




$F =$

$(x_1 \vee \overline{x_2} \vee \overline{x_3})$

$\boxed{(x_1 \vee x_2 \vee \overline{x_4})}$

$(x_1 \vee \overline{x_2} \vee \overline{x_4})$

$(x_2 \vee \overline{x_3} \vee \overline{x_4})$

$(\overline{x_1} \vee \overline{x_2} \vee x_3)$

$(\overline{x_1} \vee \overline{x_2} \vee x_4)$

$(\overline{x_1} \vee x_2 \vee x_4)$

$(\overline{x_1} \vee x_2 \vee \overline{x_4})$

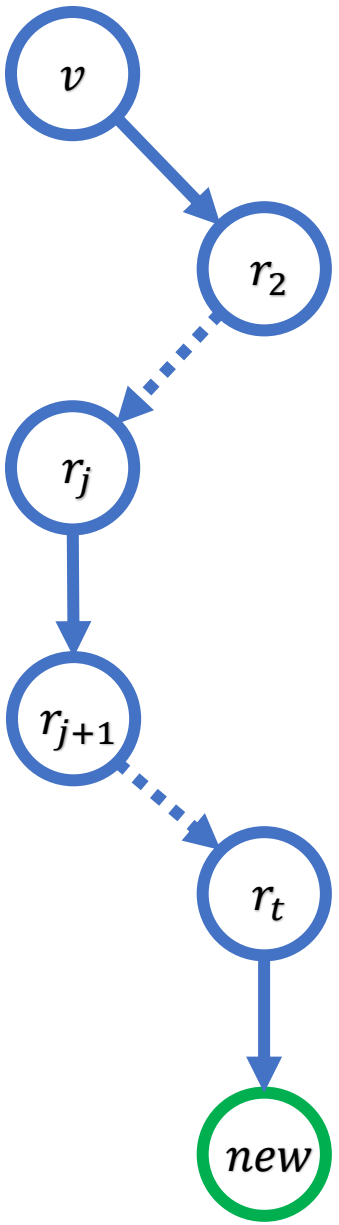
$(x_1 \vee x_2 \vee x_4)$

$(x_1 \vee x_3 \vee x_4)$

# Case 2 cont

- Let $P_i = (v, r_1, \dots, r_t)$
- By reverse induction for any $j = t, \dots, 0$
  $\exists D_j = R_j \vee S_j \vee U_j \in F_s$ s.t
  - $R_j$ is <u>negation</u> of variables from $N$
  - $S_j$ consist <u>negation</u> of variables from $A'$
  - $U_j \subseteq \{v, r_1, \dots, r_j\}$ (<u>positively</u>)

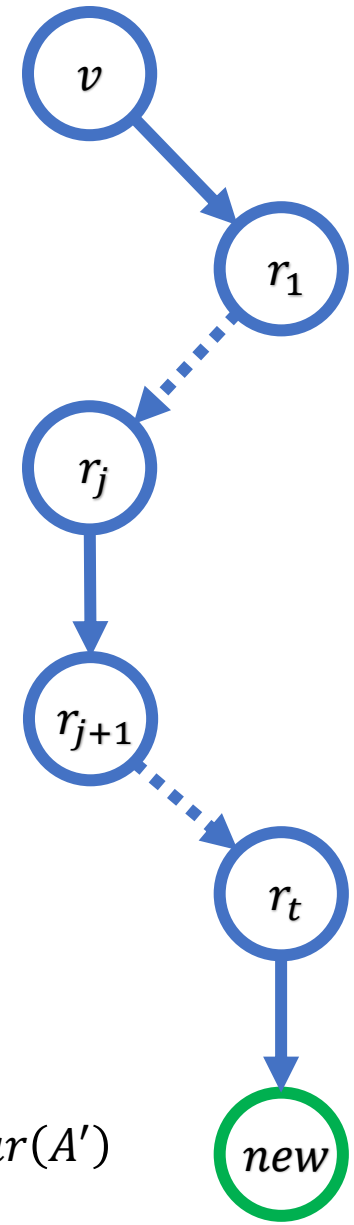
- Given the above,
  - $D_0 = R_0 \vee S_0 \vee U_0$ where $U_0 \subseteq \{v\}$
  - $U_0$ can't be empty since then $D_0$ is all <u>negation</u> of variables
  - ➢ $D_0$ is the desired critical clause



# Reverse induction proof

- **<u>Base</u> ($j = t$):** take $S_t = \emptyset$, $R_t = R$ and $U_t = U$. That is,
$$D_t = C_i$$
- **<u>Step</u>**: Assume $D_{j+1} = R_{j+1} \vee S_{j+1} \vee U_{j+1} \in F_s$
  - $R_{j+1}$ consist <u>negation</u> of variables from $N$
  - $S_{j+1}$ consist <u>negation</u> of variables from $A'$
  - $U_{j+1} \subseteq \{v, r_1, \dots, r_j, r_{j+1}\}$ (<u>positively</u>)
- If $r_{j+1}$ does not appear in $U_{j+1}$, take $D_j = D_{j+1}$
- Otherwise, $r_{j+1} \in U_{j+1}$, (w.l.o.g, $\overline{r_{j+1}} \in \boldsymbol{C(A_{j+1})}$) and the rest of $r_l$ does not appear in $\boldsymbol{C(A_{j+1})}$.
- $\boldsymbol{C(A_{j+1})}$ and $\boldsymbol{D_{j+1}}$ are resolvable
- $D_j = R(C(A_{j+1}), D_{j+1})$

All $j \in P_i$, $var(A_j) \nsubseteq var(A')$
i.e. $r_j \notin \mathrm{var}(A')$



# Using critical clause trees

- We established the **existence** of critical clause tree. We'll use that to get a lower bound on $P_v$
- We only take in consideration the critical clauses the that associated with cuts of the tree.

$$P_v \geq P(v \text{ is last in } C(A) \text{ for some cut } A \text{ in } T)$$

Where $T$ is a critical clause tree for $v$

# Using critical clause trees

**<u>Lemma 5</u>**

If there is a critical clause tree for $(v, G)$ with maximum degree $k-1$ and depth $d$, then

$$P_v \geq \frac{\mu_k}{k-1} - \epsilon_k^{(d)}$$

where $\mu_k = \sum_{j=1}^{\infty} \frac{1}{j(j+\frac{1}{k-1})}$ and $\epsilon_k^{(d)} = \frac{3}{(d-1)(k-1)+2}$

- For $k = 3$, $\mu_3 = 4 - 4\ln(2) \approx 1.226$

# Using critical clause trees

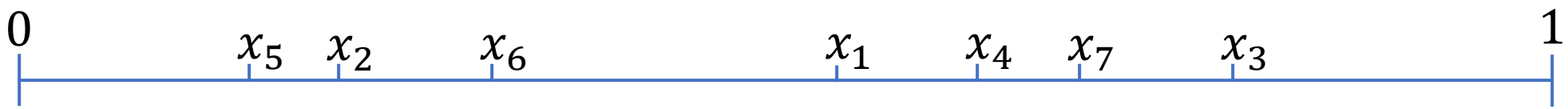
**<u>Lemma 5</u>** (for $k = 3$)
If there is a critical clause tree for $(v, G)$ with maximum degree 2 and depth $d$, then

$$P_v \geq \frac{\mu}{2} - \frac{1}{d}$$

where $\mu = 4 - 4\ln(2) \approx 1.226$

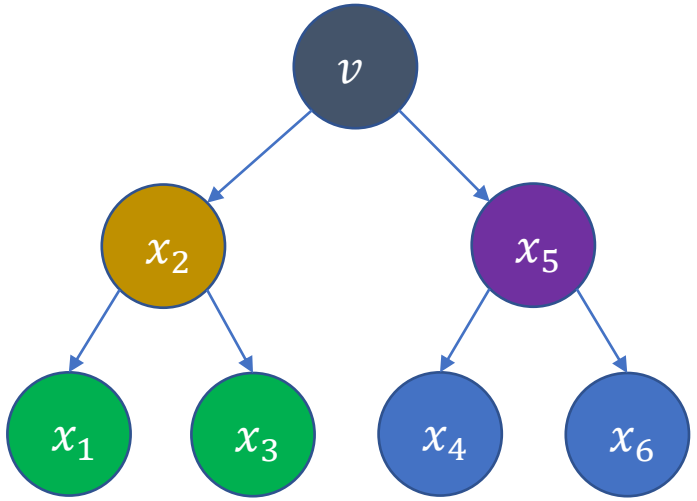
# Proof technique

- Create a random permutation in the following way:
  - Map each variable to a random value $\alpha(v) \sim U(0,1)$
  - Sort the variables with respect to $\alpha$



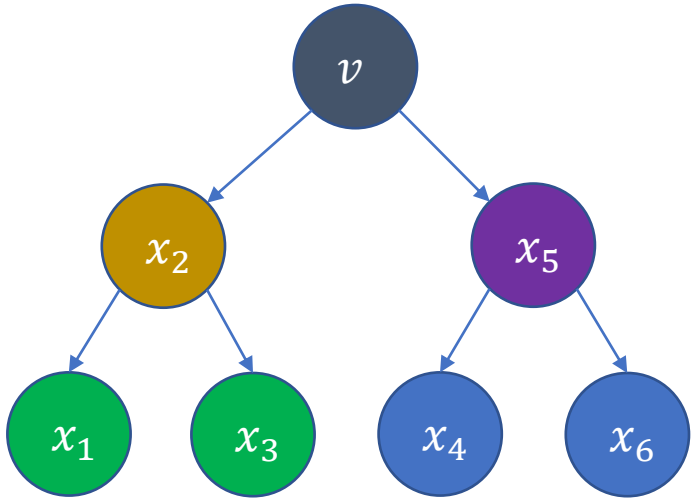
# Proof technique

- Given this tree, $v$ is forced if
  - ($\boldsymbol{x_1}$ **and** $\boldsymbol{x_3}$ are before $v$) **Or** ($\boldsymbol{x_2}$ is before $v$)
    **And** ($\boldsymbol{x_4}$ **and** $\boldsymbol{x_6}$ are before $v$) **Or** ($\boldsymbol{x_5}$ is before $v$)
  - Fix $\alpha(v) = r$, denote
    $$p_i = p(\alpha(x_i) \leq r) = r$$
    the probability for the event above, given $r$ is
    $$\big(1 - (1 - p_1 p_3)(1 - p_2)\big)\big(1 - (1 - p_4 p_6)(1 - p_5)\big)$$
    $$= (r + (1 - r)r^2)^2$$
  - Averaging over $r$, $P_v \geq \frac{31}{70}$



# Simplifications

- Each node have <u>exactly</u> $k - 1$ childrens
- All nodes are labeled
- All variables are different
- This is the "hard" case



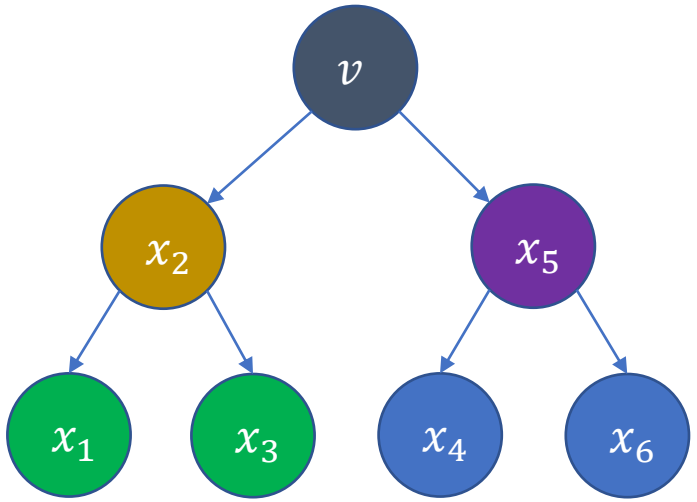
# Proof

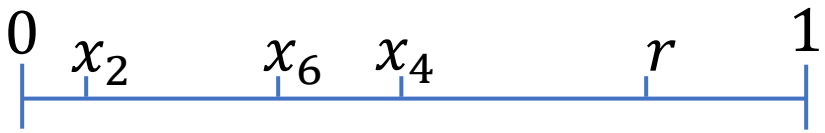
- Given admissible tree $T$ rooted at $v$, define

$$Q_T(r) := P(\exists A, \forall w \in A, \alpha(w) < r)$$

- If $T$ is a critical clause tree then

$$P_v \geq \int_0^1 Q_T(r)dr$$

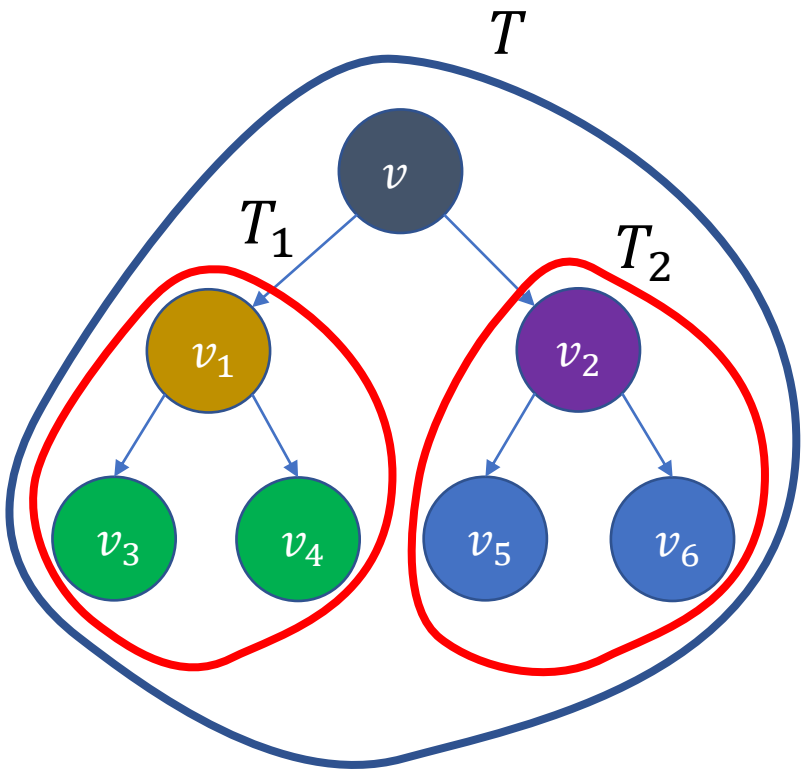


# Proof cont

- Let $T$ be an admissible tree rooted at $v$
- Let $v_1$ and $v_2$ be $v$'s children
- Let $T_1, T_2$ be the admissible sub-trees rooted $v_1, v_2$
- $E_i = (\exists A \in T_i, \forall w \in A, \alpha(w) < r)$

$$
\begin{aligned}
Q_T(r) &= P\big((\alpha(v_1) < r \vee E_1) \wedge (\alpha(v_2) < r \vee E_2)\big) \\
&= P(\alpha(v_1) < r \vee E_1) \cdot P(\alpha(v_2) < r \vee E_2) \\
&= \big(P(\alpha(v_1) < r \vee E_1)\big)^2 \\
&= \Big(r + (1-r)Q_{T_1}(r)\Big)^2
\end{aligned}
$$



# Proof cont

$$Q_d(r) = (r + (1-r)Q_{d-1}(r))^2 \qquad (Q_0 \equiv 0)$$

- What is the limit $Q_d(r)$? (as $d \to \infty$)
  - The minimal fixed point:

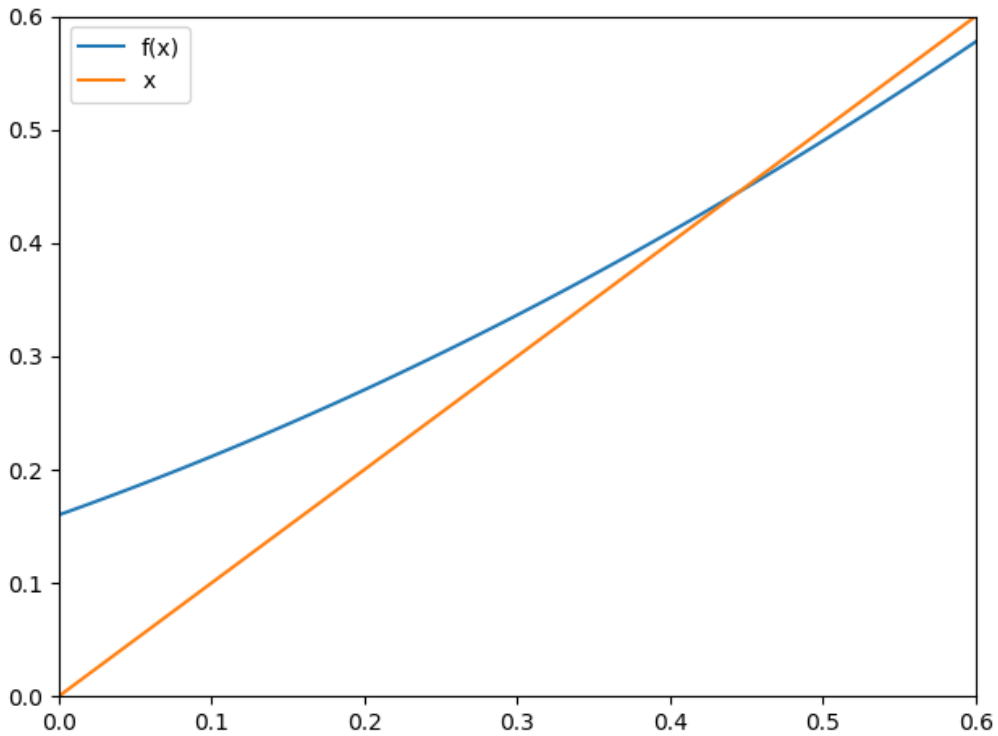
$$\min\left\{\left(\frac{r}{1-r}\right)^2, 1\right\}$$

Hence

$$\frac{\mu_3}{2} = \lim_{d\to\infty} \int_0^1 Q_d(r)\,dr = \int_0^1 \min\left\{\left(\frac{r}{1-r}\right)^2, 1\right\} dr$$

$$= 2 - 2\ln(2) \approx 0.6137$$

$$f(x) = (r + (1-r)x)^2$$



# Proof cont

- $R(r) := \min\left\{\left(\frac{r}{1-r}\right)^2, 1\right\}$
- $\mu := \int_0^1 R(r)dr$
- $\Delta_d(r) = R(r) - Q_d(r)$

$$\Delta_d(r) = R(r) - \left(r + (1-r)Q_{d-1}(r)\right)^2$$

$$= \left(r + (1-r)R(r)\right)^2 - \left(r + (1-r)\left(R(r) - \Delta_{d-1}(r)\right)\right)^2$$

$$= (1-r)\Delta_{d-1}(r)\left(2r + 2(1-r)R(r) - (1-r)\Delta_{d-1}(r)\right)$$

$$\leq 2(1-r)\left(r + (1-r)R(r)\right)\Delta_{d-1}(r)$$

# Proof cont

$$\Delta_d(r) \leq \Big(2(1-r)\big(r + (1-r)R(r)\big)\Big)^d$$

- For $0 \leq r \leq {}^1/_2$, the RHS is $(2r)^d$
- For ${}^1/_2 \leq r \leq 1$, the RHS is $(2(1-r))^d$

$$\int_0^1 \Delta_d(r) \leq \frac{1}{d+1} \leq \frac{1}{d}$$

- $\Delta_d(r) = R(r) - Q_d(r)$

$$P_v \geq \int_0^1 Q_d(r) \geq \frac{\mu}{2} - \frac{1}{d}$$

# Wrapping up

PPSZ($F$)

- $F_s = \text{Resolve}(F, s)$
- For $I$ times:
  - $\pi \sim U(S_n)$
  - $y \sim U(\{0,1\}^n)$
  - $u = Modify(F_s, \pi, y)$
  - If $u$ satisfy $F$, return $u$
- return "unsatisfiable"

$$\tau \geq 2^{-n+E_\pi[|Forced(\pi,z)|]}$$
$$= 2^{-n+\sum_v P_v}$$
$$\geq 2^{-\left(1-\frac{\mu}{2}+\frac{1}{d}\right)n}$$

- Resolve takes $O(n^s)$
- $s = \log(n)$
  - ➢Resolve takes $O\left(2^{\log^2(n)}\right)$

# Wrapping up

- To get critical clause tree of depth $d$ we need $s \geq k^d$
- For $\mathrm{s} = \log(n)$ we can get $d = \Omega(\log(\log(n)))$
- $\tau \geq 2^{-\left(1-\frac{\mu}{2}+\frac{c}{\log(\log(n))}\right)n}$
- Take $I = n2^{\left(1-\frac{\mu}{2}+\frac{c}{\log(\log(n))}\right)n}$
- Probability of success is at least $1 - e^{-n}$

- Complexity is

$$poly(n)2^{\left(1-\frac{\mu}{2}+\frac{c}{\log(\log(n))}\right)n} + 2^{o(n)}$$

$$= O^*\left(2^{0.3862n+o(n)}\right)$$

$$= O^*(1.308^n)$$

# Concluding remarks

- For unique $k$-SAT, we get complexity of

$$O^*\left(2^{\left(1-\frac{\mu_k}{k-1}\right)n}\right)$$

  - Where $\mu_k = \sum_{j=1}^{\infty} \frac{1}{j(j+\frac{1}{k-1})}$

- This can be generalized for general $k$-SAT (Hertli, 2011)

- Can be improved for unique 3-SAT (Hansen et al. 2019)

- Still an open problem if we can get better than $2^n$ for general SAT