BILL AND NATHAN, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!

ETH and NPC Probs on Graphs and Planar Graphs

Exposition by William Gasarch—U of MD

Is Assuming $\mathrm{P} \neq \mathrm{NP}$ Enough?

Exposition by William Gasarch—U of MD

What if Nathan Proved $\mathrm{P} \neq \mathrm{NP}$ in a Way that Nobody Cared?

Scenario Nathan shows $\mathrm{3SAT}$ requires $n^{\log\log\log n}$ steps.

What if Nathan Proved $\mathrm{P} \neq \mathrm{NP}$ in a Way that Nobody Cared?

Scenario Nathan shows $\mathrm{3SAT}$ requires $n^{\log\log\log n}$ steps.

PRO Nathan gets $1,000,000 and he promised that for ever $1000 he wins I get a free lunch.

What if Nathan Proved $\mathrm{P} \neq \mathrm{NP}$ in a Way that Nobody Cared?

Scenario Nathan shows $\mathrm{3SAT}$ requires $n^{\log \log \log n}$ steps.

PRO Nathan gets \$1,000,000 and he promised that for ever \$1000 he wins I get a free lunch.

CON We believe $\mathrm{3SAT}$ is a hard problem. We still don't quite know that.

What if Nathan Proved $\mathrm{P} \neq \mathrm{NP}$ in a Way that Nobody Cared?

Scenario Nathan shows $\mathrm{3SAT}$ requires $n^{\log \log \log n}$ steps.

PRO Nathan gets $1,000,000 and he promised that for ever $1000 he wins I get a free lunch.

CON We believe $\mathrm{3SAT}$ is a hard problem. We still don't quite know that.

CON For all NPC problems A there is a c such that A requires $n^{c(\log \log \log n)}$. So they have not really been proven hard.

What if Nathan Proved $\mathrm{P} \neq \mathrm{NP}$ in a Way that Nobody Cared?

Scenario Nathan shows $\mathrm{3SAT}$ requires $n^{\log\log\log n}$ steps.

PRO Nathan gets \$1,000,000 and he promised that for ever \$1000 he wins I get a free lunch.

CON We believe $\mathrm{3SAT}$ is a hard problem. We still don't quite know that.

CON For all NPC problems A there is a c such that A requires $n^{c(\log\log\log n)}$. So they have not really been proven hard.

So... What should Nathan try to prove now?

We Believe More than $\mathrm{P} \neq \mathrm{NP}$

So we think $\mathrm{3SAT} \notin \mathrm{P}$. Whats the best known algorithm?

We Believe More than $\mathrm{P} \neq \mathrm{NP}$

So we think $\mathrm{3SAT} \notin \mathrm{P}$. Whats the best known algorithm?

1. $\mathrm{3SAT}$ is in deterministic time $O(1.3303^n) \sim 2^{0.418n}$.

We Believe More than $\mathrm{P} \neq \mathrm{NP}$

So we think $\mathrm{3SAT} \notin \mathrm{P}$. Whats the best known algorithm?

1. $\mathrm{3SAT}$ is in deterministic time $O(1.3303^n) \sim 2^{0.418n}$.
2. $\mathrm{3SAT}$ is in randomized time $O(1.308^n) \sim 2^{0.387n}$.

How much better can we get?

We Believe More than $\mathrm{P} \neq \mathrm{NP}$

So we think $3\mathrm{SAT} \notin \mathrm{P}$. Whats the best known algorithm?

1. $3\mathrm{SAT}$ is in deterministic time $O(1.3303^n) \sim 2^{0.418n}$.
2. $3\mathrm{SAT}$ is in randomized time $O(1.308^n) \sim 2^{0.387n}$.

How much better can we get?

1. It is plausible that there is an algorithm in (say) $2^{0.1n}$.

We Believe More than $\mathrm{P} \neq \mathrm{NP}$

So we think $3\mathrm{SAT} \notin \mathrm{P}$. Whats the best known algorithm?

1. $3\mathrm{SAT}$ is in deterministic time $O(1.3303^n) \sim 2^{0.418n}$.
2. $3\mathrm{SAT}$ is in randomized time $O(1.308^n) \sim 2^{0.387n}$.

How much better can we get?

1. It is plausible that there is an algorithm in (say) $2^{0.1n}$.
2. It is unlikely that there is an algorithm in (say) $n^{(\log n)^{10}}$.

We Believe More than $\mathrm{P} \neq \mathrm{NP}$

So we think $3\mathrm{SAT} \notin \mathrm{P}$. Whats the best known algorithm?

1. $3\mathrm{SAT}$ is in deterministic time $O(1.3303^n) \sim 2^{0.418n}$.
2. $3\mathrm{SAT}$ is in randomized time $O(1.308^n) \sim 2^{0.387n}$.

How much better can we get?

1. It is plausible that there is an algorithm in (say) $2^{0.1n}$.
2. It is unlikely that there is an algorithm in (say) $n^{(\log n)^{10}}$.

Hypotheses The **Exponential Time Hypothesis (ETH)** is that $3\mathrm{SAT}$ requires $2^{\Omega(n)}$ steps.

We Believe More than $\mathrm{P} \neq \mathrm{NP}$

So we think $\mathrm{3SAT} \notin \mathrm{P}$. Whats the best known algorithm?

1. $\mathrm{3SAT}$ is in deterministic time $O(1.3303^n) \sim 2^{0.418n}$.
2. $\mathrm{3SAT}$ is in randomized time $O(1.308^n) \sim 2^{0.387n}$.

How much better can we get?

1. It is plausible that there is an algorithm in (say) $2^{0.1n}$.
2. It is unlikely that there is an algorithm in (say) $n^{(\log n)^{10}}$.

Hypotheses The **Exponential Time Hypothesis (ETH)** is that $\mathrm{3SAT}$ requires $2^{\Omega(n)}$ steps.
Note If ETH is true this gives a good lower bound on NPC problems. How good? We will encounter this later.

We Believe More than $\mathrm{P} \neq \mathrm{NP}$

So we think $\mathrm{3SAT} \notin \mathrm{P}$. Whats the best known algorithm?

1. $\mathrm{3SAT}$ is in deterministic time $O(1.3303^n) \sim 2^{0.418n}$.
2. $\mathrm{3SAT}$ is in randomized time $O(1.308^n) \sim 2^{0.387n}$.

How much better can we get?

1. It is plausible that there is an algorithm in (say) $2^{0.1n}$.
2. It is unlikely that there is an algorithm in (say) $n^{(\log n)^{10}}$.

Hypotheses The **Exponential Time Hypothesis (ETH)** is that $\mathrm{3SAT}$ requires $2^{\Omega(n)}$ steps.
Note If ETH is true this gives a good lower bound on NPC problems. How good? We will encounter this later.
Nathan Go to it!

One Subtle Point about ETH

ETH is in terms of n, the number of **variables**.

One Subtle Point about ETH

ETH is in terms of n, the number of **variables**.

What about the number of **clauses**?
Our reductions may depend on then number of **clauses**.

One Subtle Point about ETH

ETH is in terms of n, the number of **variables**.

What about the number of **clauses**?
Our reductions may depend on then number of **clauses**.

We will revisit this point after the next reduction.

NPC Problems on Graphs

Exposition by William Gasarch—U of MD

The Clique Problem

Def If G is a graph then a **clique** is a set of vertices such that every pair has an edge.

The Clique Problem

Def If G is a graph then a **clique** is a set of vertices such that every pair has an edge.

Def

$$\mathrm{CLIQ} = \{(G, k) : G \text{ has a clique of size } k \}.$$

We show that CLIQ is NPC.

3SAT $\leq$ CLIQ

1) Input $\phi = C_1 \wedge \cdots \wedge C_k$ where each C_i is a 3-clause.

$3\mathrm{SAT} \le \mathrm{CLIQ}$

1) Input $\phi = C_1 \wedge \cdots \wedge C_k$ where each C_i is a 3-clause.

2) Graph G with $7k$ vertices as follows: For each clause we have 7 vertices. Label them with the 7 ways to set the 3 vars to make the clause satisfiable. For example, for the clause $x \vee y \vee \neg z$, we have 7 vertices: TTT, TTF, TFT, TFF, FTT, FTF, FFF,

$\mathbf{3SAT} \le \mathbf{CLIQ}$

1) Input $\phi = C_1 \wedge \cdots \wedge C_k$ where each C_i is a 3-clause.

2) Graph G with $7k$ vertices as follows: For each clause we have 7 vertices. Label them with the 7 ways to set the 3 vars to make the clause satisfiable. For example, for the clause $x \vee y \vee \neg z$, we have 7 vertices: TTT, TTF, TFT, TFF, FTT, FTF, FFF,

There are no edges between vertices associated to the same clause. We put an edge between vertices associated with different clauses if the assignments do not conflict. Example:
$(x = T, y = T, z = T)$ has edge to $(w = F, x = T, z = T)$ but not to $(w = F, x = F, z = T)$.

$3\mathrm{SAT} \le \mathrm{CLIQ}$

1) Input $\phi = C_1 \wedge \cdots \wedge C_k$ where each C_i is a 3-clause.

2) Graph G with $7k$ vertices as follows: For each clause we have 7 vertices. Label them with the 7 ways to set the 3 vars to make the clause satisfiable. For example, for the clause $x \vee y \vee \neg z$, we have 7 vertices: TTT, TTF, TFT, TFF, FTT, FTF, FFF,

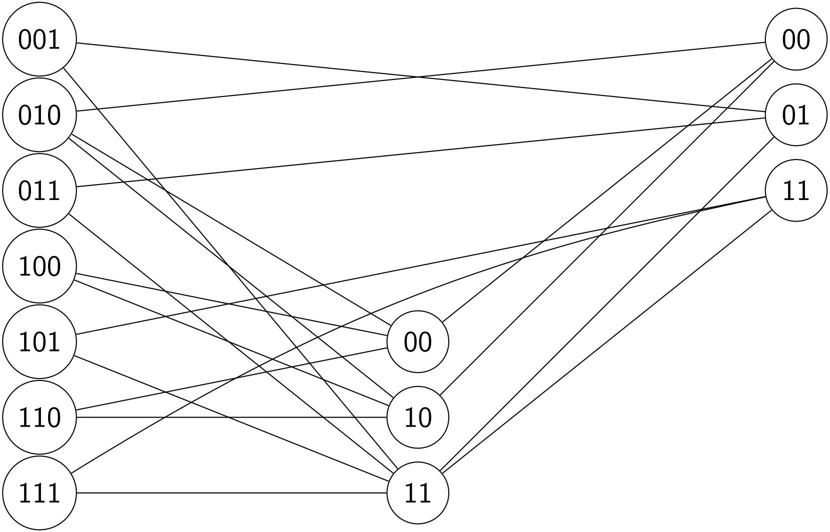
There are no edges between vertices associated to the same clause. We put an edge between vertices associated with different clauses if the assignments do not conflict. Example:
$(x = T, y = T, z = T)$ has edge to $(w = F, x = T, z = T)$ but not to $(w = F, x = F, z = T)$.

3) Example on next slide

BILL AND NATHAN RECORD

BILL AND NATHAN RECORD

$(x \vee y \vee z) \quad \wedge \quad (w \vee \overline{z}) \quad \wedge \quad (\overline{x} \vee z)$



There is More to the Story of $\mathrm{3SAT} \le \mathrm{CLIQ}$

Notation v is numb of verts since n is numb of vars.
Since we believe $\mathrm{3SAT} \notin \mathrm{P}$, we believe $\mathrm{CLIQ} \notin \mathrm{P}$.

There is More to the Story of $\mathrm{3SAT} \le \mathrm{CLIQ}$

Notation v is numb of verts since n is numb of vars.
Since we believe $\mathrm{3SAT} \notin \mathrm{P}$, we believe $\mathrm{CLIQ} \notin \mathrm{P}$.

But we believe ETH: $\mathrm{3SAT}$ requires $2^{\Omega(n)}$ time.

There is More to the Story of $3\mathrm{SAT} \le \mathrm{CLIQ}$

Notation v is numb of verts since n is numb of vars.
Since we believe $3\mathrm{SAT} \notin \mathrm{P}$, we believe $\mathrm{CLIQ} \notin \mathrm{P}$.

But we believe ETH: $3\mathrm{SAT}$ requires $2^{\Omega(n)}$ time.

Mapped ϕ to a graph G with $\le 7m$ vertices.

There is More to the Story of $3\mathrm{SAT} \le \mathrm{CLIQ}$

Notation v is numb of verts since n is numb of vars.
Since we believe $3\mathrm{SAT} \notin \mathrm{P}$, we believe $\mathrm{CLIQ} \notin \mathrm{P}$.

But we believe ETH: $3\mathrm{SAT}$ requires $2^{\Omega(n)}$ time.

Mapped ϕ to a graph G with $\le 7m$ vertices.

Want $3\mathrm{SAT}$ requires $2^{\Omega(n)} \rightarrow \mathrm{CLIQ}$ requires $2^{\Omega(v)}$.

There is More to the Story of $\mathrm{3SAT} \le \mathrm{CLIQ}$

Notation v is numb of verts since n is numb of vars.
Since we believe $\mathrm{3SAT} \notin \mathrm{P}$, we believe $\mathrm{CLIQ} \notin \mathrm{P}$.

But we believe ETH: $\mathrm{3SAT}$ requires $2^{\Omega(n)}$ time.

Mapped ϕ to a graph G with $\le 7m$ vertices.

Want $\mathrm{3SAT}$ requires $2^{\Omega(n)} \rightarrow \mathrm{CLIQ}$ requires $2^{\Omega(v)}$.

Don't Have We need a linear reduction. We have ϕ on n variables, m clauses maps to graph with $7m$ clauses.

There is More to the Story of $\mathrm{3SAT} \le \mathrm{CLIQ}$

Notation v is numb of verts since n is numb of vars.
Since we believe $\mathrm{3SAT} \notin \mathrm{P}$, we believe $\mathrm{CLIQ} \notin \mathrm{P}$.

But we believe ETH: $\mathrm{3SAT}$ requires $2^{\Omega(n)}$ time.

Mapped ϕ to a graph G with $\le 7m$ vertices.

Want $\mathrm{3SAT}$ requires $2^{\Omega(n)} \rightarrow \mathrm{CLIQ}$ requires $2^{\Omega(v)}$.

Don't Have We need a linear reduction. We have ϕ on n variables, m clauses maps to graph with $7m$ clauses.

Revisit ETH on next slide.

What Does ETH Mean?

What does **$\mathbf{3SAT}$ requires $2^{\Omega(n)}$ steps** mean?

What Does ETH Mean?

What does **$\mathrm{3SAT}$ requires $2^{\Omega(n)}$ steps** mean?
It means $(\exists c > 0)[\mathrm{3SAT} \notin \mathrm{DTIME}(2^{cn})]$.

What Does ETH Mean?

What does **3SAT requires $2^{\Omega(n)}$ steps** mean?
It means $(\exists c > 0)[\mathrm{3SAT} \notin \mathrm{DTIME}(2^{cn})]$.

Let A be a graph problem.
What does ***A* requires $2^{\Omega(v)}$ steps** mean?

What Does ETH Mean?

What does **$\mathrm{3SAT}$ requires $2^{\Omega(n)}$ steps** mean?
It means $(\exists c > 0)[\mathrm{3SAT} \notin \mathrm{DTIME}(2^{cn})]$.

Let A be a graph problem.
What does ***A* requires $2^{\Omega(v)}$ steps** mean?
It means $(\exists d > 0)[A \notin \mathrm{DTIME}(2^{dv})]$.

What Does ETH Mean?

What does **$\mathrm{3SAT}$ requires $2^{\Omega(n)}$ steps** mean?
It means $(\exists c > 0)[\mathrm{3SAT} \notin \mathrm{DTIME}(2^{cn})]$.

Let A be a graph problem.
What does ***A* requires $2^{\Omega(v)}$ steps** mean?
It means $(\exists d > 0)[A \notin \mathrm{DTIME}(2^{dv})]$.

We want
$(\exists c > 0)[\mathrm{3SAT} \notin \mathrm{DTIME}(2^{cn})] \rightarrow (\exists d > 0)[A \notin \mathrm{DTIME}(2^{dv})]$.

What Does ETH Mean?

What does **$3\mathrm{SAT}$ requires $2^{\Omega(n)}$ steps** mean?
It means $(\exists c > 0)[3\mathrm{SAT} \notin \mathrm{DTIME}(2^{cn})]$.

Let A be a graph problem.
What does ***A* requires $2^{\Omega(v)}$ steps** mean?
It means $(\exists d > 0)[A \notin \mathrm{DTIME}(2^{dv})]$.

We want
$(\exists c > 0)[3\mathrm{SAT} \notin \mathrm{DTIME}(2^{cn})] \rightarrow (\exists d > 0)[A \notin \mathrm{DTIME}(2^{dv})]$.

Easier to proof contrapositive
$(\forall d > 0)[A \in \mathrm{DTIME}(2^{dv})] \rightarrow (\forall c > 0)[3\mathrm{SAT} \in \mathrm{DTIME}(2^{cn})]$.

What Does ETH Mean?

What does **$\mathrm{3SAT}$ requires $2^{\Omega(n)}$ steps** mean?
It means $(\exists c > 0)[\mathrm{3SAT} \notin \mathrm{DTIME}(2^{cn})]$.

Let A be a graph problem.
What does ***A* requires $2^{\Omega(v)}$ steps** mean?
It means $(\exists d > 0)[A \notin \mathrm{DTIME}(2^{dv})]$.

We want
$(\exists c > 0)[\mathrm{3SAT} \notin \mathrm{DTIME}(2^{cn})] \rightarrow (\exists d > 0)[A \notin \mathrm{DTIME}(2^{dv})]$.

Easier to proof contrapositive
$(\forall d > 0)[A \in \mathrm{DTIME}(2^{dv})] \rightarrow (\forall c > 0)[\mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.

This is equivalent to
$(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$ then A requires $2^{\Omega(v)}$.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$
then A requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$ then A requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Pick a c. We pick d later.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$ then A requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Pick a c. We pick d later.
1) Input ϕ on n variables.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$ then A requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Pick a c. We pick d later.
1) Input ϕ on n variables.
2) Compute $f(\phi)$ to get a graph G with $v \le an$ vertices. **Time** poly in n, so negligible, we ignore.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$ then A requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Pick a c. We pick d later.
1) Input ϕ on n variables.
2) Compute $f(\phi)$ to get a graph G with $v \le an$ vertices. **Time** poly in n, so negligible, we ignore.
3) Run 2^{dv} alg for A on G. (We chose d later.) **Time** $\le 2^{adn}$.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$ then A requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Pick a c. We pick d later.
1) Input ϕ on n variables.
2) Compute $f(\phi)$ to get a graph G with $v \le an$ vertices. **Time** poly in n, so negligible, we ignore.
3) Run 2^{dv} alg for A on G. (We chose d later.) **Time** $\le 2^{adn}$.
4) Ouput the answer.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$ then A requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Pick a c. We pick d later.
1) Input ϕ on n variables.
2) Compute $f(\phi)$ to get a graph G with $v \le an$ vertices. **Time** poly in n, so negligible, we ignore.
3) Run 2^{dv} alg for A on G. (We chose d later.) **Time** $\le 2^{adn}$.
4) Ouput the answer.
This takes time 2^{adn}. Take $d = \frac{c}{a}$ to get a 2^{cn} alg for $\mathrm{3SAT}$.

How to use ETH to Get $2^{\Omega(v)}$ Lower Bounds

Let A be a graph problem. We want to show that A takes $2^{\Omega(v)}$.
Thm (ETH) $\mathrm{3SAT} \le A$ by reduction f AND $(\exists a)[|f(\phi)| \le an]$ then A requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[A \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Pick a c. We pick d later.
1) Input ϕ on n variables.
2) Compute $f(\phi)$ to get a graph G with $v \le an$ vertices. **Time** poly in n, so negligible, we ignore.
3) Run 2^{dv} alg for A on G. (We chose d later.) **Time** $\le 2^{adn}$.
4) Ouput the answer.
This takes time 2^{adn}. Take $d = \frac{c}{a}$ to get a 2^{cn} alg for $\mathrm{3SAT}$.
Note If the reduction took $2^{\epsilon n}$ time, would still work. If needed to do the reduction many times would still work.

Recap of Our Dilemma

We have $\mathrm{3SAT} \le \mathrm{CLIQ}$ where a formula on n variables and m clauses maps to a graph on $\le 7m$ vertices.
So cannot use ETH to show that CLIQ requires $2^{\Omega(v)}$.

Recap of Our Dilemma

We have $\mathrm{3SAT} \le \mathrm{CLIQ}$ where a formula on n variables and m clauses maps to a graph on $\le 7m$ vertices.
So cannot use ETH to show that CLIQ requires $2^{\Omega(v)}$.

We need a way to reduce $\mathrm{3SAT}$ problems to a $\mathrm{3SAT}$ problem where the number of clauses is $\le O(n)$.

Recap of Our Dilemma

We have $\mathrm{3SAT} \le \mathrm{CLIQ}$ where a formula on n variables and m clauses maps to a graph on $\le 7m$ vertices.
So cannot use ETH to show that CLIQ requires $2^{\Omega(v)}$.

We need a way to reduce $\mathrm{3SAT}$ problems to a $\mathrm{3SAT}$ problem where the number of clauses is $\le O(n)$.

Since we are dealing with exponential time the reduction will take exponential time but in a controlled way.

Recap of Our Dilemma

We have $3\mathrm{SAT} \le \mathrm{CLIQ}$ where a formula on n variables and m clauses maps to a graph on $\le 7m$ vertices.
So cannot use ETH to show that CLIQ requires $2^{\Omega(v)}$.

We need a way to reduce $3\mathrm{SAT}$ problems to a $3\mathrm{SAT}$ problem where the number of clauses is $\le O(n)$.

Since we are dealing with exponential time the reduction will take exponential time but in a controlled way.

Notation $\tilde{O}(f(n))$: ignore polys. Often used as $\tilde{O}(2^{cn})$.
ETH is equiv to: $(\exists c)(\forall p(n))$ $3\mathrm{SAT}$ requires $\ge p(n)2^{\Omega(cn)}$.

Sparsification Lemma (SL)

For all $\epsilon > 0$ there exists a constant $e(\epsilon)$ and a $\tilde{O}(2^{\epsilon n})$ algorithm that does the following:

Input ϕ, $\mathrm{3CNF}$ formula.

Sparsification Lemma (SL)

For all $\epsilon > 0$ there exists a constant $e(\epsilon)$ and a $\tilde{O}(2^{\epsilon n})$ algorithm that does the following:

Input ϕ, $3\mathrm{CNF}$ formula.

Output $3\mathrm{CNF}$ formulas $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:

Sparsification Lemma (SL)

For all $\epsilon > 0$ there exists a constant $e(\epsilon)$ and a $\tilde{O}(2^{\epsilon n})$ algorithm that does the following:

Input ϕ, $3\mathrm{CNF}$ formula.

Output $3\mathrm{CNF}$ formulas $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:

(1) Each ϕ_i has $\leq e(\epsilon)n$ clauses

Sparsification Lemma (SL)

For all $\epsilon > 0$ there exists a constant $e(\epsilon)$ and a $\tilde{O}(2^{\epsilon n})$ algorithm that does the following:

Input ϕ, $\mathrm{3CNF}$ formula.

Output $\mathrm{3CNF}$ formulas $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:

(1) Each ϕ_i has $\leq e(\epsilon)n$ clauses

(2) $\phi \in \mathrm{3SAT}$ iff $(\exists i)[\phi_i \in \mathrm{3SAT}]$.

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use?

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!
Pick a c. We pick ϵ and d later.

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!
Pick a c. We pick ϵ and d later.

1. Input ϕ in $\mathrm{3CNF}$, on n variables.

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!
Pick a c. We pick ϵ and d later.

1. Input ϕ in $\mathrm{3CNF}$, on n variables.
2. SL: In time $\tilde{O}(2^{\epsilon n})$ get $\mathrm{3CNF}$ $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!
Pick a c. We pick ϵ and d later.

1. Input ϕ in $\mathrm{3CNF}$, on n variables.
2. SL: In time $\tilde{O}(2^{\epsilon n})$ get $\mathrm{3CNF}$ $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:
   (1) Each ϕ_i has $\leq e(\epsilon)n$ clauses.

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!
Pick a c. We pick ϵ and d later.

1. Input ϕ in $\mathrm{3CNF}$, on n variables.
2. SL: In time $\tilde{O}(2^{\epsilon n})$ get $\mathrm{3CNF}$ $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:
   (1) Each ϕ_i has $\le e(\epsilon)n$ clauses.
   (2) $\phi \in \mathrm{3SAT}$ iff $(\exists i)[\phi_i \in \mathrm{3SAT}]$.

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!
Pick a c. We pick ϵ and d later.

1. Input ϕ in $\mathrm{3CNF}$, on n variables.
2. SL: In time $\tilde{O}(2^{\epsilon n})$ get $\mathrm{3CNF}$ $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:
   (1) Each ϕ_i has $\le e(\epsilon)n$ clauses.
   (2) $\phi \in \mathrm{3SAT}$ iff $(\exists i)[\phi_i \in \mathrm{3SAT}]$.
3. For each ϕ_i do reduction, get (G_i, k_i). **Key** ϕ_i has $\le e(\epsilon)n$ clauses, so G_i has $\le 7e(\epsilon)n$ vertices. This took time $\tilde{O}(2^{\epsilon n})$.

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!
Pick a c. We pick ϵ and d later.

1. Input ϕ in $\mathrm{3CNF}$, on n variables.
2. SL: In time $\tilde{O}(2^{\epsilon n})$ get $\mathrm{3CNF}$ $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:
   (1) Each ϕ_i has $\le e(\epsilon)n$ clauses.
   (2) $\phi \in \mathrm{3SAT}$ iff $(\exists i)[\phi_i \in \mathrm{3SAT}]$.
3. For each ϕ_i do reduction, get (G_i, k_i). **Key** ϕ_i has $\le e(\epsilon)n$ clauses, so G_i has $\le 7e(\epsilon)n$ vertices. This took time $\tilde{O}(2^{\epsilon n})$.
4. Do the 2^{dv} algorithm on each (G_i, k_i). This takes time

$$\tilde{O}(2^{\epsilon n} \times 2^{7de(\epsilon))n}) = \tilde{O}(2^{(\epsilon+7de(\epsilon))n}).$$

Using the SL

Thm (ETH) CLIQ requires $2^{\Omega(v)}$.
Need $(\forall c)(\exists d)[\mathrm{CLIQ} \in \mathrm{DTIME}(2^{dv}) \rightarrow \mathrm{3SAT} \in \mathrm{DTIME}(2^{cn})]$.
Erika What Lemma do we use? Right! SL!
Pick a c. We pick ϵ and d later.

1. Input ϕ in $\mathrm{3CNF}$, on n variables.
2. SL: In time $\tilde{O}(2^{\epsilon n})$ get $\mathrm{3CNF}$ $\phi_1, \ldots, \phi_{2^{\epsilon n}}$ such that:
   (1) Each ϕ_i has $\le e(\epsilon)n$ clauses.
   (2) $\phi \in \mathrm{3SAT}$ iff $(\exists i)[\phi_i \in \mathrm{3SAT}]$.
3. For each ϕ_i do reduction, get (G_i, k_i). **Key** ϕ_i has $\le e(\epsilon)n$ clauses, so G_i has $\le 7e(\epsilon)n$ vertices. This took time $\tilde{O}(2^{\epsilon n})$.
4. Do the 2^{dv} algorithm on each (G_i, k_i). This takes time

$$\tilde{O}(2^{\epsilon n} \times 2^{7de(\epsilon))n}) = \tilde{O}(2^{(\epsilon+7de(\epsilon))n}).$$

   Pick d, ϵ on next slide.

ETH implies CLIQ Requires $2^{\Omega(v)}$

Given c we want to pick d, ϵ such tha t

$$2^{(\epsilon+7de(\epsilon))n} < 2^{cn}$$

ETH implies CLIQ Requires $2^{\Omega(v)}$

Given c we want to pick d, ϵ such tha t

$$2^{(\epsilon+7de(\epsilon))n} < 2^{cn}$$

$$(\epsilon + 7de(\epsilon))n < cn$$

ETH implies CLIQ Requires $2^{\Omega(v)}$

Given c we want to pick d, ϵ such tha t

$$2^{(\epsilon+7de(\epsilon))n} < 2^{cn}$$

$$(\epsilon + 7de(\epsilon))n < cn$$

$$\epsilon + 7de(\epsilon) < c$$

ETH implies CLIQ Requires $2^{\Omega(v)}$

Given c we want to pick d, ϵ such tha t

$$2^{(\epsilon+7de(\epsilon))n} < 2^{cn}$$

$$(\epsilon + 7de(\epsilon))n < cn$$

$$\epsilon + 7de(\epsilon) < c$$

Pick $\epsilon = \frac{c}{3}$.

ETH implies CLIQ Requires $2^{\Omega(v)}$

Given c we want to pick d, ϵ such tha t

$$2^{(\epsilon+7de(\epsilon))n} < 2^{cn}$$

$$(\epsilon + 7de(\epsilon))n < cn$$

$$\epsilon + 7de(\epsilon) < c$$

Pick $\epsilon = \frac{c}{3}$.
Pick $d = \frac{c}{14e(\epsilon)}$.

ETH implies CLIQ Requires $2^{\Omega(v)}$

Given c we want to pick d, ϵ such tha t

$$2^{(\epsilon+7de(\epsilon))n} < 2^{cn}$$

$$(\epsilon + 7de(\epsilon))n < cn$$

$$\epsilon + 7de(\epsilon) < c$$

Pick $\epsilon = \frac{c}{3}$.
Pick $d = \frac{c}{14e(\epsilon)}$.
Then

$$\epsilon + 7de(\epsilon) = \frac{c}{3} + \frac{c}{2} = \frac{5c}{6} < c.$$

Recap

From

Recap

From

1. ETH

Recap

From

1. ETH
2. $3\mathrm{SAT} \leq \mathrm{CLIQ}$ is linear.

Recap

From

1. ETH
2. $3\mathrm{SAT} \leq \mathrm{CLIQ}$ is linear.
3. SL.

Recap

From

1. ETH
2. $\mathrm{3SAT} \le \mathrm{CLIQ}$ is linear.
3. SL.

we get that CLIQ requires $2^{\Omega(v)}$.

Recap

From

1. ETH
2. $\mathrm{3SAT} \le \mathrm{CLIQ}$ is linear.
3. SL.

we get that CLIQ requires $2^{\Omega(v)}$.

A similar argument can be made for all of the graph problems in the rest of this talk.

Recap

From

1. ETH
2. $\mathrm{3SAT} \le \mathrm{CLIQ}$ is linear.
3. SL.

we get that CLIQ requires $2^{\Omega(v)}$.

A similar argument can be made for all of the graph problems in the rest of this talk.

So we won't bother.

The Ind. Set Problem

Def If G is a graph then an **ind. set** is a set of vertices such that no pair has an edge.

The Ind. Set Problem

Def If G is a graph then an **ind. set** is a set of vertices such that no pair has an edge.

Def

$$\mathrm{IS} = \{(G, k) : G \text{ has an ind. set of size } k\ \}.$$

We show that IS is NPC.

Two ways to prove IS is NPC

Two ways to prove IS is NPC

Method One Prove $3\mathrm{SAT} \leq \mathrm{IS}$, similar to $3\mathrm{SAT} \leq \mathrm{CLIQ}$.

Two ways to prove IS is NPC

Method One Prove $3\mathrm{SAT} \leq \mathrm{IS}$, similar to $3\mathrm{SAT} \leq \mathrm{CLIQ}$.
Method Two Show $\mathrm{CLIQ} \leq \mathrm{IS}$.

Two ways to prove IS is NPC

Method One Prove $\mathrm{3SAT} \le \mathrm{IS}$, similar to $\mathrm{3SAT} \le \mathrm{CLIQ}$.
Method Two Show $\mathrm{CLIQ} \le \mathrm{IS}$.
Easy $(G, k) \in \mathrm{CLIQ}$ iff $(\overline{G}, k) \in \mathrm{IS}$.
($\overline{G}$ is $(V, \binom{V}{2} - E)$.)

Two ways to prove IS is NPC

Method One Prove $3\mathrm{SAT} \le \mathrm{IS}$, similar to $3\mathrm{SAT} \le \mathrm{CLIQ}$.
Method Two Show $\mathrm{CLIQ} \le \mathrm{IS}$.
Easy $(G, k) \in \mathrm{CLIQ}$ iff $(\overline{G}, k) \in \mathrm{IS}$.
($\overline{G}$ is $(V, \binom{V}{2} - E)$.)
Moral Once you have many NPC sets you can use them rather than SAT. In the future we won't bother with method one.

Two ways to prove IS is NPC

Method One Prove $3\mathrm{SAT} \le \mathrm{IS}$, similar to $3\mathrm{SAT} \le \mathrm{CLIQ}$.
Method Two Show $\mathrm{CLIQ} \le \mathrm{IS}$.
Easy $(G, k) \in \mathrm{CLIQ}$ iff $(\overline{G}, k) \in \mathrm{IS}$.
($\overline{G}$ is $(V, \binom{V}{2} - E)$.)
Moral Once you have many NPC sets you can use them rather than SAT. In the future we won't bother with method one.
Bonus Reduction is linear, so assuming ETH and using SL we have IS requires $2^{\Omega(v)}$ time.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def

$$\mathrm{VC} = \{(G,k) : G \text{ has a vertex cover of size } k\ \}.$$

We show that VC is NPC.

VC is NPC

We show $\mathrm{IS} \leq \mathrm{VC}$.

VC is NPC

We show $\mathrm{IS} \le \mathrm{VC}$.
GO TO BREAKOUT ROOMS TO TRY TO PROVE THIS.

VC is NPC

We show $\mathrm{IS} \le \mathrm{VC}$.
GO TO BREAKOUT ROOMS TO TRY TO PROVE THIS.
$(G, k) \in \mathrm{IS}$ iff $(G, n-k) \in \mathrm{VC}$.
I leave the proof to you.

VC is NPC

We show $\mathrm{IS} \le \mathrm{VC}$.
GO TO BREAKOUT ROOMS TO TRY TO PROVE THIS.
$(G,k) \in \mathrm{IS}$ iff $(G, n-k) \in \mathrm{VC}$.
I leave the proof to you.
Bonus Reduction is linear, so assuming ETH and using SL, VC requires $2^{\Omega(v)}$ time.

RECAP

RECAP

1. CLIQ is NPC.

RECAP

1. CLIQ is NPC.
2. IS is NPC.

RECAP

1. CLIQ is NPC.
2. IS is NPC.
3. VC is NPC.

RECAP

1. CLIQ is NPC.
2. IS is NPC.
3. VC is NPC.
4. All of the reductions were linear so, assuming ETH, and using the sparsification lemma, all three problems requires $2^{\Omega(v)}$ time.

RECAP

1. CLIQ is NPC.
2. IS is NPC.
3. VC is NPC.
4. All of the reductions were linear so, assuming ETH, and using the sparsification lemma, all three problems requires $2^{\Omega(v)}$ time.

We now look at graph coloring.

Graph Coloring

Def A graph is k-colorable if can map vertices to $\{1, \ldots, k\}$ such that no adjacent vertices are the same color.

Graph Coloring

Def A graph is k-colorable if can map vertices to $\{1, \dots, k\}$ such that no adjacent vertices are the same color.

$$\mathrm{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

Graph Coloring

Def A graph is k-colorable if can map vertices to $\{1, \ldots, k\}$ such that no adjacent vertices are the same color.

$$\mathrm{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

$k = 1$ is in P but silly, G is 1-col iff G has no edges.

Graph Coloring

Def A graph is k-colorable if can map vertices to $\{1, \ldots, k\}$ such that no adjacent vertices are the same color.

$$\mathrm{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

$k = 1$ is in P but silly, G is 1-col iff G has no edges.
$k = 2$ is in P: Color 1 RED and everything is forced.

Graph Coloring

Def A graph is k-colorable if can map vertices to $\{1, \ldots, k\}$ such that no adjacent vertices are the same color.

$$\mathrm{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

$k = 1$ is in P but silly, G is 1-col iff G has no edges.
$k = 2$ is in P: Color 1 RED and everything is forced.

$k = 3$ is first interesting case. We show $\mathrm{3COL}$ is NPC.

Graph Coloring

Def A graph is k-colorable if can map vertices to $\{1, \ldots, k\}$ such that no adjacent vertices are the same color.

$$\mathrm{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

$k = 1$ is in P but silly, G is 1-col iff G has no edges.
$k = 2$ is in P: Color 1 RED and everything is forced.

$k = 3$ is first interesting case. We show $3\mathrm{COL}$ is NPC.

Fractional Colorings One can define fractional colorings, so a graph can be $\frac{5}{2}$-colorable. For all $k > 2$, kCOL is NPC. We won't define or prove this.

3COL is NPC

Given $\phi = C_1 \vee \cdots \vee C_k$ in 3-CNF form we produce G such that

$$\phi \in \mathrm{3SAT} \text{ iff } G \in \mathrm{3COL}$$

3COL is NPC

Given $\phi = C_1 \vee \cdots \vee C_k$ in 3-CNF form we produce G such that

$$\phi \in \mathrm{3SAT} \text{ iff } G \in \mathrm{3COL}$$

We construct three gadgets on the next three slides.

3COL is NPC

Given $\phi = C_1 \vee \cdots \vee C_k$ in 3-CNF form we produce G such that

$$\phi \in \mathrm{3SAT} \text{ iff } G \in \mathrm{3COL}$$

We construct three gadgets on the next three slides.

The first two gadgets will set things up. They have nothing to do with the formula except for the number-of-variables.

3COL is NPC

Given $\phi = C_1 \vee \cdots \vee C_k$ in 3-CNF form we produce G such that

$$\phi \in \mathrm{3SAT} \text{ iff } G \in \mathrm{3COL}$$

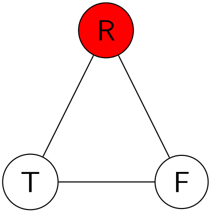
We construct three gadgets on the next three slides.

The first two gadgets will set things up. They have nothing to do with the formula except for the number-of-variables.

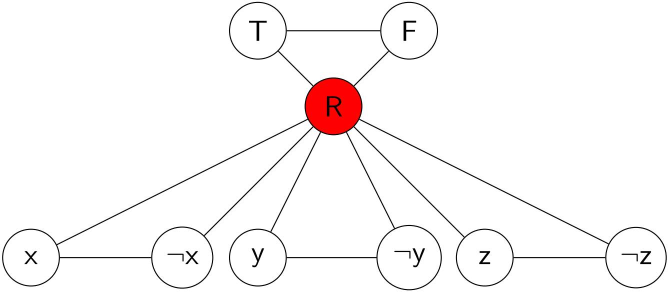
The third gadget involves the clauses.

TRUE, FALSE, and RED

We have the following triangle. The colors are *not* part of the graph; however,
we will think of later when a variable is colored T (F) then it is set to T (F).
We will make sure that no variable is colored R



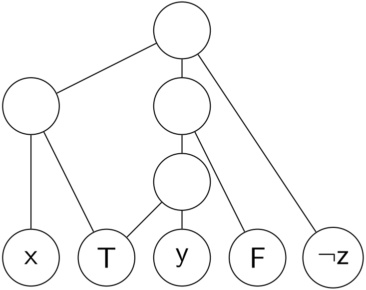
No var is R. $(x, \overline{x})$ is (T,F) or (F,T)



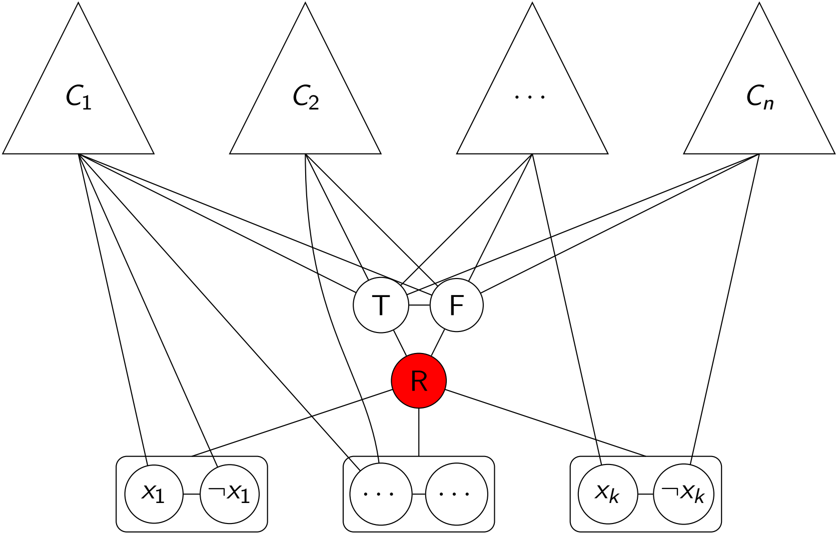
The Clause Gadget for $x \vee y \vee \neg z$

Recall that $x, y, \neg z$ are colored T or F.

1. If $x, y, \neg z$ are all colored F then NOT 3-colorable.
2. If $x, y, \neg z$ are anything else, then IS 3-colorable.



Putting it All Together



ETH and 3COL

You can check the reduction gives G of size $O(n+m)$.

ETH and $3\mathrm{COL}$

You can check the reduction gives G of size $O(n+m)$.
ETH Using ETH and SL $3\mathrm{COL}$ requires $2^{\Omega(v)}$.

4-COL, 5-COL, ... all NPC

Erika How to show that, for $k \geq 3$, kCOL is NPC?

4-COL, 5-COL, . . . all NPC

Erika How to show that, for $k \geq 3$, kCOL is NPC?
By a reduction.

4-COL, 5-COL, ... all NPC

Erika How to show that, for $k \geq 3$, kCOL is NPC?
By a reduction.

Anyone Give me the reduction $3\mathrm{COL} \leq 4\mathrm{COL}$.

4-COL, 5-COL, ... all NPC

Erika How to show that, for $k \geq 3$, kCOL is NPC?
By a reduction.

Anyone Give me the reduction $\mathrm{3COL} \leq \mathrm{4COL}$.

Anyone Give me the reduction $\mathrm{3COL} \leq \mathrm{kCOL}$.

4-COL, 5-COL, . . . all NPC

Erika How to show that, for $k \geq 3$, kCOL is NPC?
By a reduction.

Anyone Give me the reduction $3\mathrm{COL} \leq 4\mathrm{COL}$.

Anyone Give me the reduction $3\mathrm{COL} \leq \mathrm{kCOL}$.

Question Is $4\mathrm{COL} \leq 3\mathrm{COL}$? Vote

4-COL, 5-COL, . . . all NPC

Erika How to show that, for $k \geq 3$, kCOL is NPC?
By a reduction.

Anyone Give me the reduction $3\mathrm{COL} \leq 4\mathrm{COL}$.

Anyone Give me the reduction $3\mathrm{COL} \leq \mathrm{kCOL}$.

Question Is $4\mathrm{COL} \leq 3\mathrm{COL}$? Vote

YES

4-COL, 5-COL, ... all NPC

Erika How to show that, for $k \geq 3$, kCOL is NPC?
By a reduction.

Anyone Give me the reduction $3\mathrm{COL} \leq 4\mathrm{COL}$.

Anyone Give me the reduction $3\mathrm{COL} \leq \mathrm{kCOL}$.

Question Is $4\mathrm{COL} \leq 3\mathrm{COL}$? Vote

YES but by an insane reduction:

$$4\mathrm{COL} \leq 3\mathrm{SAT} \leq 3\mathrm{COL}.$$

4-COL, 5-COL, . . . all NPC

Erika How to show that, for $k \geq 3$, kCOL is NPC?
By a reduction.

Anyone Give me the reduction $3\mathrm{COL} \leq 4\mathrm{COL}$.

Anyone Give me the reduction $3\mathrm{COL} \leq \mathrm{kCOL}$.

Question Is $4\mathrm{COL} \leq 3\mathrm{COL}$? Vote

YES but by an insane reduction:

$$4\mathrm{COL} \leq 3\mathrm{SAT} \leq 3\mathrm{COL}.$$

Is there a sane reduction?

4-COL, 5-COL, ... all NPC

Erika How to show that, for $k \ge 3$, kCOL is NPC?
By a reduction.

Anyone Give me the reduction $3\mathrm{COL} \le 4\mathrm{COL}$.

Anyone Give me the reduction $3\mathrm{COL} \le \mathrm{kCOL}$.

Question Is $4\mathrm{COL} \le 3\mathrm{COL}$? Vote

YES but by an insane reduction:

$$4\mathrm{COL} \le 3\mathrm{SAT} \le 3\mathrm{COL}.$$

Is there a sane reduction? Yes. Tell story about it.

Sanity

We give the Sane Reduction by first giving two Gadgets.

Gadget 1: x, y both $c \to z$ is c

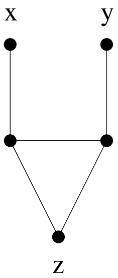
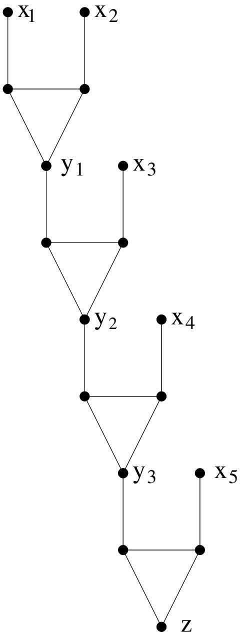


Figure: $\mathrm{GAD}(x, y, z)$

Gadget 2: x_1, x_2, x_3, x_4, x_5 all $c \rightarrow z$ is c



$\mathbf{4COL} \le \mathbf{3SAT}$

Given $G = (V, E)$ we want to construct $G' = (V', E')$ such that G is 4-Colorable iff G' is 3-Colorable

$\mathbf{4COL} \leq \mathbf{3SAT}$

Given $G = (V, E)$ we want to construct $G' = (V', E')$ such that

$$G \text{ is 4-Colorable iff } G' \text{ is 3-Colorable}$$

Create a graph G' as follows:

SET UP This will not involve G at all except for the number of vertices.

For every $i \in V$ and $j \in \{1, 2, 3, 4\}$ have node v_{ij}.

4COL $\leq$ 3SAT

Given $G = (V, E)$ we want to construct $G' = (V', E')$ such that

$$G \text{ is 4-Colorable iff } G' \text{ is 3-Colorable}$$

Create a graph G' as follows:

SET UP This will not involve G at all except for the number of vertices.

For every $i \in V$ and $j \in \{1, 2, 3, 4\}$ have node v_{ij}.

Intent:

If v_{ij} is colored T in G' then i is colored j in G.

If v_{ij} is colored F in G' then i is NOT colored j in G.

$4\mathrm{COL} \le 3\mathrm{SAT}$

Given $G = (V, E)$ we want to construct $G' = (V', E')$ such that

G is 4-Colorable iff G' is 3-Colorable

Create a graph G' as follows:

SET UP This will not involve G at all except for the number of vertices.

For every $i \in V$ and $j \in \{1, 2, 3, 4\}$ have node v_{ij}.

Intent:

If v_{ij} is colored T in G' then i is colored j in G.

If v_{ij} is colored F in G' then i is NOT colored j in G.

We use the tri-gadget to make sure that v_{ij} is not R.

$4\mathrm{COL} \leq 3\mathrm{SAT}$

Given $G = (V, E)$ we want to construct $G' = (V', E')$ such that

G is 4-Colorable iff G' is 3-Colorable

Create a graph G' as follows:

SET UP This will not involve G at all except for the number of vertices.

For every $i \in V$ and $j \in \{1, 2, 3, 4\}$ have node v_{ij}.

Intent:

If v_{ij} is colored T in G' then i is colored j in G.
If v_{ij} is colored F in G' then i is NOT colored j in G.
We use the tri-gadget to make sure that v_{ij} is not R.

Need to make sure that, for all $i \in V$:

1. At least one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T
2. At most one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T

Will do this on next slide.

4COL $\leq$ 3SAT SET UP

Fix i.

$4\mathrm{COL} \le 3\mathrm{SAT}$ SET UP

Fix i.

1. Have $\mathrm{GAD}(v_{i1}, v_{i2}, v_{i3}, v_{i4}, T)$.

$4\mathrm{COL} \le 3\mathrm{SAT}$ SET UP

Fix i.

1. Have $\mathrm{GAD}(v_{i1}, v_{i2}, v_{i3}, v_{i4}, T)$.
   This ensures that at least one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T

$4\mathrm{COL} \le 3\mathrm{SAT}$ SET UP

Fix i.

1. Have $\mathrm{GAD}(v_{i1}, v_{i2}, v_{i3}, v_{i4}, T)$.
   This ensures that at least one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T
2. Have
   $\mathrm{GAD}(v_{i1}, v_{i2}, F)$,
   $\mathrm{GAD}(v_{i1}, v_{i3}, F)$,
   $\mathrm{GAD}(v_{i1}, v_{i4}, F)$,
   $\mathrm{GAD}(v_{i2}, v_{i3}, F)$,
   $\mathrm{GAD}(v_{i2}, v_{i4}, F)$,
   $\mathrm{GAD}(v_{i3}, v_{i4}, F)$.

$4\mathrm{COL} \le 3\mathrm{SAT}$ SET UP

Fix i.

1. Have $\mathrm{GAD}(v_{i1}, v_{i2}, v_{i3}, v_{i4}, T)$.
   This ensures that at least one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T
2. Have
   $\mathrm{GAD}(v_{i1}, v_{i2}, F)$,
   $\mathrm{GAD}(v_{i1}, v_{i3}, F)$,
   $\mathrm{GAD}(v_{i1}, v_{i4}, F)$,
   $\mathrm{GAD}(v_{i2}, v_{i3}, F)$,
   $\mathrm{GAD}(v_{i2}, v_{i4}, F)$,
   $\mathrm{GAD}(v_{i3}, v_{i4}, F)$.
   This ensures that at most one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T

$4\mathrm{COL} \le 3\mathrm{SAT}$ SET UP

Fix i.

1. Have $\mathrm{GAD}(v_{i1}, v_{i2}, v_{i3}, v_{i4}, T)$.
   This ensures that at least one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T
2. Have
   $\mathrm{GAD}(v_{i1}, v_{i2}, F)$,
   $\mathrm{GAD}(v_{i1}, v_{i3}, F)$,
   $\mathrm{GAD}(v_{i1}, v_{i4}, F)$,
   $\mathrm{GAD}(v_{i2}, v_{i3}, F)$,
   $\mathrm{GAD}(v_{i2}, v_{i4}, F)$,
   $\mathrm{GAD}(v_{i3}, v_{i4}, F)$.
   This ensures that at most one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T

Have: any 3-coloring (proper or not) of G' will induce a 4-coloring (proper or not) on G.

$4\mathrm{COL} \le 3\mathrm{SAT}$ SET UP

Fix i.

1. Have $\mathrm{GAD}(v_{i1}, v_{i2}, v_{i3}, v_{i4}, T)$.
   This ensures that at least one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T
2. Have
   $\mathrm{GAD}(v_{i1}, v_{i2}, F)$,
   $\mathrm{GAD}(v_{i1}, v_{i3}, F)$,
   $\mathrm{GAD}(v_{i1}, v_{i4}, F)$,
   $\mathrm{GAD}(v_{i2}, v_{i3}, F)$,
   $\mathrm{GAD}(v_{i2}, v_{i4}, F)$,
   $\mathrm{GAD}(v_{i3}, v_{i4}, F)$.
   This ensures that at most one of $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ is T

Have: any 3-coloring (proper or not) of G' will induce a 4-coloring (proper or not) on G.

Need: if G' has a proper 3-coloring then the induced 4-coloring is proper.

$4COL \leq 3SAT$ The Heart of the Construction

Need to make sure that two adjacent vertices of G have different colors.

$4\mathrm{COL} \le 3\mathrm{SAT}$ The Heart of the Construction

Need to make sure that two adjacent vertices of G have different colors.

For all $(i,j) \in E$ we have the following gadgets.
$\mathrm{GAD}(v_{i1}, v_{j1}, F)$,
$\mathrm{GAD}(v_{i2}, v_{j2}, F)$,
$\mathrm{GAD}(v_{i3}, v_{j3}, F)$,
$\mathrm{GAD}(v_{i4}, v_{j4}, F)$.

$4\mathrm{COL} \le 3\mathrm{SAT}$ The Heart of the Construction

Need to make sure that two adjacent vertices of G have different colors.

For all $(i,j) \in E$ we have the following gadgets.
$\mathrm{GAD}(v_{i1}, v_{j1}, F)$,
$\mathrm{GAD}(v_{i2}, v_{j2}, F)$,
$\mathrm{GAD}(v_{i3}, v_{j3}, F)$,
$\mathrm{GAD}(v_{i4}, v_{j4}, F)$.

Gee, that wasn't hard at all! The SETUP was most of the construction!

$4\mathrm{COL} \le 3\mathrm{SAT}$ The Heart of the Construction

Need to make sure that two adjacent vertices of G have different colors.

For all $(i, j) \in E$ we have the following gadgets.
$\mathrm{GAD}(v_{i1}, v_{j1}, F)$,
$\mathrm{GAD}(v_{i2}, v_{j2}, F)$,
$\mathrm{GAD}(v_{i3}, v_{j3}, F)$,
$\mathrm{GAD}(v_{i4}, v_{j4}, F)$.

Gee, that wasn't hard at all! The SETUP was most of the construction!

Similar proof shows $\mathrm{kCOL} \le 3\mathrm{COL}$.

NPC Problems on Planar Graphs

Exposition by William Gasarch—U of MD

Restrict to Planar Graphs

We look at the graph problems we just proved NPC and see what happens when restricted to planar graphs.

Restrict to Planar Graphs

We look at the graph problems we just proved NPC and see what happens when restricted to planar graphs.

Testing Planarity is in P so we can assume the graph given IS Planar.

Clique on Planar Graphs

CLIQ: One of the following is true

1. CLIQ restricted to Planar graphs is NPC.
2. CLIQ restricted to Planar graphs is in P.
3. The status of CLIQ restricted to Planar graphs is unknown to science.

Vote

Clique on Planar Graphs

CLIQ: One of the following is true

1. CLIQ restricted to Planar graphs is NPC.
2. CLIQ restricted to Planar graphs is in P.
3. The status of CLIQ restricted to Planar graphs is unknown to science.

Vote

CLIQ restricted to Planar graph is in P.

Clique on Planar Graphs

CLIQ: One of the following is true

1. CLIQ restricted to Planar graphs is NPC.
2. CLIQ restricted to Planar graphs is in P.
3. The status of CLIQ restricted to Planar graphs is unknown to science.

Vote
CLIQ restricted to Planar graph is in P.
If G is planar then G DOES NOT have a clique of size ≥ 5.

Clique on Planar Graphs

CLIQ: One of the following is true

1. CLIQ restricted to Planar graphs is NPC.
2. CLIQ restricted to Planar graphs is in P.
3. The status of CLIQ restricted to Planar graphs is unknown to science.

Vote
CLIQ restricted to Planar graph is in P.
If G is planar then G DOES NOT have a clique of size ≥ 5.

$$\{(G, k) : G \text{ is Planar and } G \text{ has a clique of size } k\}$$

Clique on Planar Graphs

CLIQ: One of the following is true

1. CLIQ restricted to Planar graphs is NPC.
2. CLIQ restricted to Planar graphs is in P.
3. The status of CLIQ restricted to Planar graphs is unknown to science.

Vote
CLIQ restricted to Planar graph is in P.
If G is planar then G DOES NOT have a clique of size ≥ 5.

$$\{(G, k) : G \text{ is Planar and } G \text{ has a clique of size } k\}$$

If $k \geq 5$ just say NO.

Clique on Planar Graphs

CLIQ: One of the following is true

1. CLIQ restricted to Planar graphs is NPC.
2. CLIQ restricted to Planar graphs is in P.
3. The status of CLIQ restricted to Planar graphs is unknown to science.

Vote

CLIQ restricted to Planar graph is in P.

If G is planar then G DOES NOT have a clique of size ≥ 5.

$$\{(G, k) : G \text{ is Planar and } G \text{ has a clique of size } k\}$$

If $k \geq 5$ just say NO.

If $k \leq 4$ can do brute force in $O(n^k) \leq O(n^4)$ time.

Ind Set and Vertex Cover on Planar Graphs

One of the following is true:

1. IS restricted to Planar graphs is NPC.
2. IS restricted to Planar graphs is in P.
3. The status of IS restricted to Planar graphs is unknown to science.

Vote

Ind Set and Vertex Cover on Planar Graphs

One of the following is true:

1. IS restricted to Planar graphs is NPC.
2. IS restricted to Planar graphs is in P.
3. The status of IS restricted to Planar graphs is unknown to science.

Vote IS restricted to Planar graph is in NPC.

Ind Set and Vertex Cover on Planar Graphs

One of the following is true:

1. IS restricted to Planar graphs is NPC.
2. IS restricted to Planar graphs is in P.
3. The status of IS restricted to Planar graphs is unknown to science.

Vote IS restricted to Planar graph is in NPC.
There is a reduction from IS to planar-IS but we won't be using it.

Ind Set and Vertex Cover on Planar Graphs

One of the following is true:

1. IS restricted to Planar graphs is NPC.
2. IS restricted to Planar graphs is in P.
3. The status of IS restricted to Planar graphs is unknown to science.

Vote IS restricted to Planar graph is in NPC.
There is a reduction from IS to planar-IS but we won't be using it.
We will do something else later.

NPC Coloring of Graphs and Planar Graphs

Exposition by William Gasarch—U of MD

Coloring of Graphs and Planar Graphs

$$\mathrm{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

$$\mathrm{PL\text{-}kCOL} = \{G : G \text{ is planar and } k\text{-colorable}\}$$

In breakout rooms discuss planar $3\mathrm{COL}$, $4\mathrm{COL}$, $5\mathrm{COL}$, $\ldots$.

Coloring of Graphs and Planar Graphs

$$\mathrm{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

$$\mathrm{PL\text{-}kCOL} = \{G : G \text{ is planar and } k\text{-colorable}\}$$

In breakout rooms discuss planar $3\mathrm{COL}$, $4\mathrm{COL}$, $5\mathrm{COL}$, $\ldots$.
$\mathrm{PL\text{-}3COL}$ is NPC and we will show that.

Coloring of Graphs and Planar Graphs

$$\text{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

$$\text{PL-kCOL} = \{G : G \text{ is planar and } k\text{-colorable}\}$$

In breakout rooms discuss planar 3COL, 4COL, 5COL, $\ldots$.
PL-3COL is NPC and we will show that.
PL-4COL is in P: It is known that **every planar graph is 4-colorable**.

Coloring of Graphs and Planar Graphs

$$\mathrm{kCOL} = \{G : G \text{ is } k\text{-colorable}\}$$

$$\mathrm{PL\text{-}kCOL} = \{G : G \text{ is planar and } k\text{-colorable}\}$$

In breakout rooms discuss planar $3\mathrm{COL}$, $4\mathrm{COL}$, $5\mathrm{COL}$, ...
$\mathrm{PL\text{-}3COL}$ is NPC and we will show that.
$\mathrm{PL\text{-}4COL}$ is in P: It is known that **every planar graph is 4-colorable**.
Hence for all $k \geq 4$, $\mathrm{PL\text{-}kCOL}$ is in P.

PL-3COL is NPC

Erika Ask the question and answer it.

PL-3COL is NPC

Erika Ask the question and answer it.
Yes we will prove this by a reduction.

PL-3COL is NPC

Erika Ask the question and answer it.
Yes we will prove this by a reduction.
We show $\mathrm{3COL} \le \mathrm{PL\text{-}3COL}$.

PL-3COL is NPC

Erika Ask the question and answer it.
Yes we will prove this by a reduction.
We show $3\mathrm{COL} \le \mathrm{PL\text{-}3COL}$.

Key A **crossover Gadget**

PL-3COL is NPC

Erika Ask the question and answer it.
Yes we will prove this by a reduction.
We show $3\mathrm{COL} \le \mathrm{PL\text{-}3COL}$.

Key A **crossover Gadget**

For every crossing we remove it and put in a planar gadget that has the same affect.

PL-3COL is NPC

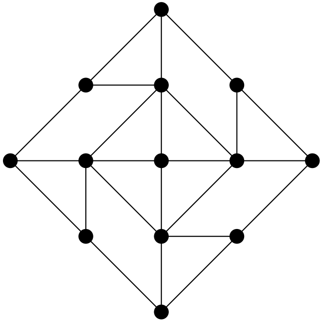
Erika Ask the question and answer it.
Yes we will prove this by a reduction.
We show $\mathrm{3COL} \le \mathrm{PL\text{-}3COL}$.

Key A **crossover Gadget**

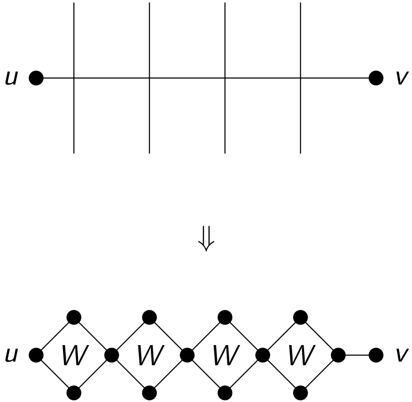
For every crossing we remove it and put in a planar gadget that has the same affect.

Gadget is on next slide and is all we need for the proof.

Crossover Gadget

$W =$ 

How to Use Crossover Gadget



ETH and $3\mathrm{COL}$

Recall:
Assume ETH. Then $3\mathrm{COL}$ requires $2^{\Omega(v)}$ time.

ETH and $3\mathrm{COL}$

Recall:
Assume ETH. Then $3\mathrm{COL}$ requires $2^{\Omega(v)}$ time.
What about $\mathrm{PL}\text{-}3\mathrm{COL}$? Discuss.

ETH and $3\mathrm{COL}$

Recall:
Assume ETH. Then $3\mathrm{COL}$ requires $2^{\Omega(v)}$ time.
What about $\mathrm{PL}\text{-}3\mathrm{COL}$? Discuss.

1. We replace all crossings with $O(1)$ vertices. Hence we care about the number of crossings.

ETH and $3\mathrm{COL}$

Recall:
Assume ETH. Then $3\mathrm{COL}$ requires $2^{\Omega(v)}$ time.
What about $\mathrm{PL}\text{-}3\mathrm{COL}$? Discuss.

1. We replace all crossings with $O(1)$ vertices. Hence we care about the number of crossings.
2. A graph with n vertices might have A LOT of crossings.

ETH and 3COL

Recall:
Assume ETH. Then $\mathrm{3COL}$ requires $2^{\Omega(v)}$ time.
What about $\mathrm{PL\text{-}3COL}$? Discuss.

1. We replace all crossings with $O(1)$ vertices. Hence we care about the number of crossings.
2. A graph with n vertices might have A LOT of crossings.
3. **Bad News** K_n probably requires $\Omega(n^4)$ crossings.

ETH and $\mathrm{3COL}$

Recall:
Assume ETH. Then $\mathrm{3COL}$ requires $2^{\Omega(v)}$ time.
What about $\mathrm{PL\text{-}3COL}$? Discuss.

1. We replace all crossings with $O(1)$ vertices. Hence we care about the number of crossings.
2. A graph with n vertices might have A LOT of crossings.
3. **Bad News** K_n probably requires $\Omega(n^4)$ crossings.
4. **Good News** We are not dealing with graphs anywhere near as complicated as K_n.

ETH and $3\mathrm{COL}$ (Cont)

ETH and $3\mathrm{COL}$ (Cont)

1. **Good News** We are only dealing with graphs from $3\mathrm{SAT} \le 3\mathrm{COL}$ reduction.

ETH and 3COL (Cont)

1. **Good News** We are only dealing with graphs from $3\mathrm{SAT} \le 3\mathrm{COL}$ reduction.
2. $3\mathrm{SAT} \le 3\mathrm{COL} \le \mathrm{PL\text{-}3COL}$. $\phi \rightarrow G \rightarrow G'$.
   ϕ n vars $\rightarrow$ G $O(n)$ verts $\rightarrow$ G' $O(n^2)$ verts.

ETH and $\mathrm{3COL}$ (Cont)

1. **Good News** We are only dealing with graphs from $\mathrm{3SAT} \le \mathrm{3COL}$ reduction.
2. $\mathrm{3SAT} \le \mathrm{3COL} \le \mathrm{PL\text{-}3COL}$. $\phi \rightarrow G \rightarrow G'$.
   ϕ n vars $\rightarrow$ G $O(n)$ verts $\rightarrow$ G' $O(n^2)$ verts.

Want lower bound on $\mathrm{PL\text{-}3COL}$ assuming ETH. Discuss.

ETH and $3\mathrm{COL}$ (Cont)

1. **Good News** We are only dealing with graphs from $3\mathrm{SAT} \le 3\mathrm{COL}$ reduction.
2. $3\mathrm{SAT} \le 3\mathrm{COL} \le \mathrm{PL\text{-}3COL}$. $\phi \to G \to G'$.
   ϕ n vars $\to$ G $O(n)$ verts $\to$ G' $O(n^2)$ verts.

Want lower bound on $\mathrm{PL\text{-}3COL}$ assuming ETH. Discuss.
Thm (ETH) $\mathrm{PL\text{-}3COL}$ requires $2^{\Omega(\sqrt{v})}$.

ETH and $3\mathrm{COL}$ (Cont)

1. **Good News** We are only dealing with graphs from $3\mathrm{SAT} \le 3\mathrm{COL}$ reduction.
2. $3\mathrm{SAT} \le 3\mathrm{COL} \le \mathrm{PL\text{-}3COL}$. $\phi \rightarrow G \rightarrow G'$.
   ϕ n vars $\rightarrow$ G $O(n)$ verts $\rightarrow$ G' $O(n^2)$ verts.

Want lower bound on $\mathrm{PL\text{-}3COL}$ assuming ETH. Discuss.
Thm (ETH) $\mathrm{PL\text{-}3COL}$ requires $2^{\Omega(\sqrt{v})}$.

Can we improve the lower bound on $\mathrm{PL\text{-}3COL}$ to $2^{\Omega(v)}$ assuming ETH?

ETH and 3COL (Cont)

1. **Good News** We are only dealing with graphs from $\mathrm{3SAT} \le \mathrm{3COL}$ reduction.
2. $\mathrm{3SAT} \le \mathrm{3COL} \le \mathrm{PL\text{-}3COL}$. $\phi \rightarrow G \rightarrow G'$.
   ϕ n vars $\rightarrow$ G $O(n)$ verts $\rightarrow$ G' $O(n^2)$ verts.

Want lower bound on $\mathrm{PL\text{-}3COL}$ assuming ETH. Discuss.
Thm (ETH) $\mathrm{PL\text{-}3COL}$ requires $2^{\Omega(\sqrt{v})}$.

Can we improve the lower bound on $\mathrm{PL\text{-}3COL}$ to $2^{\Omega(v)}$ assuming ETH? Assuming Hayes Hypothesis? **Vote!**

ETH and $3\mathrm{COL}$ (Cont)

1. **Good News** We are only dealing with graphs from $3\mathrm{SAT} \le 3\mathrm{COL}$ reduction.
2. $3\mathrm{SAT} \le 3\mathrm{COL} \le \mathrm{PL\text{-}3COL}$. $\phi \rightarrow G \rightarrow G'$.
   ϕ n vars $\rightarrow$ G $O(n)$ verts $\rightarrow$ G' $O(n^2)$ verts.

Want lower bound on $\mathrm{PL\text{-}3COL}$ assuming ETH. Discuss.
Thm (ETH) $\mathrm{PL\text{-}3COL}$ requires $2^{\Omega(\sqrt{v})}$.

Can we improve the lower bound on $\mathrm{PL\text{-}3COL}$ to $2^{\Omega(v)}$ assuming ETH? Assuming Hayes Hypothesis? **Vote!**

No There **is** a $2^{O(\sqrt{v})}$ algorithm for $\mathrm{PL\text{-}3COL}$. Comes from work on graphs of bounded treewidth.

Do We Really Want to Devise a New Crossover Gadgets For Every Problem?

Exposition by William Gasarch—U of MD

The Following Problems are NPC

The Following Problems are NPC

1. Planar Ind Set (uses a crossover gadget).

The Following Problems are NPC

1. Planar Ind Set (uses a crossover gadget).
2. Planar Vertex Cover (easily from planar ind Set).

The Following Problems are NPC

1. Planar Ind Set (uses a crossover gadget).
2. Planar Vertex Cover (easily from planar ind Set).
3. Planar Dominating Set (reduce Planar VC to it).

The Following Problems are NPC

1. Planar Ind Set (uses a crossover gadget).
2. Planar Vertex Cover (easily from planar ind Set).
3. Planar Dominating Set (reduce Planar VC to it).
4. Planar Hamiltonian Cycle (provably impossible to prove using a crossover gadget, answering a question of Gasarch).

The Following Problems are NPC

1. Planar Ind Set (uses a crossover gadget).
2. Planar Vertex Cover (easily from planar ind Set).
3. Planar Dominating Set (reduce Planar VC to it).
4. Planar Hamiltonian Cycle (provably impossible to prove using a crossover gadget, answering a question of Gasarch).
5. Others as well.

The Following Problems are NPC

1. Planar Ind Set (uses a crossover gadget).
2. Planar Vertex Cover (easily from planar ind Set).
3. Planar Dominating Set (reduce Planar VC to it).
4. Planar Hamiltonian Cycle (provably impossible to prove using a crossover gadget, answering a question of Gasarch).
5. Others as well.

Rather than do all of these proofs and devise different crossover gadgets, we define a planar variant of **SAT**.

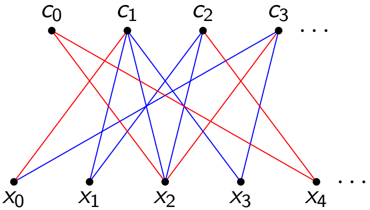
The Following Problems are NPC

1. Planar Ind Set (uses a crossover gadget).
2. Planar Vertex Cover (easily from planar ind Set).
3. Planar Dominating Set (reduce Planar VC to it).
4. Planar Hamiltonian Cycle (provably impossible to prove using a crossover gadget, answering a question of Gasarch).
5. Others as well.

Rather than do all of these proofs and devise different crossover gadgets, we define a planar variant of **SAT**.

But SAT is not a graph problem!

Graph Associated to a CNF Formula



$$c_0 = (\neg x_2 \wedge \neg x_4)$$
$$c_1 = (\neg x_0 \wedge x_1 \wedge x_2 \wedge x_3)$$
$$c_2 = (x_1 \wedge x_2 \wedge \neg x_4)$$
$$c_3 = (x_0 \wedge \neg x_2 \wedge x_3)$$
$$\cdots$$

Figure: Bipartite Graph Associated to a CNF Formula

Planar SAT

Def $\mathrm{PL\text{-}3SAT}$: Given a 3CNF formula ϕ, whose graph is planar, is it satisfiable?

Planar SAT

Def $\mathrm{PL\text{-}3SAT}$: Given a 3CNF formula ϕ, whose graph is planar, is it satisfiable?

1. $\mathrm{PL\text{-}3SAT}$ is NPC. This will require a crossover gadget.

Planar SAT

Def $\mathrm{PL\text{-}3SAT}$: Given a 3CNF formula ϕ, whose graph is planar, is it satisfiable?

1. $\mathrm{PL\text{-}3SAT}$ is NPC. This will require a crossover gadget.
2. For several graph problems X we will give reductions
   $\mathrm{3SAT} \le X$ that
   map ϕ to G, and
   if ϕ is planar then G is planar.

Planar SAT

Def $\mathrm{PL\text{-}3SAT}$: Given a 3CNF formula ϕ, whose graph is planar, is it satisfiable?

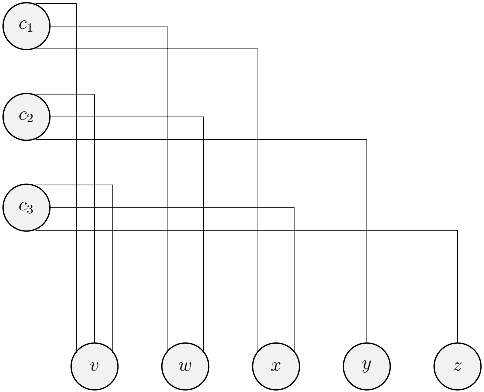
1. $\mathrm{PL\text{-}3SAT}$ is NPC. This will require a crossover gadget.
2. For several graph problems X we will give reductions
   $\mathrm{3SAT} \le X$ that
   map ϕ to G, and
   if ϕ is planar then G is planar.
3. Hence we will show many planar graph problems NPC without having to construct a gadget for each one.

PL-3SAT is NPC

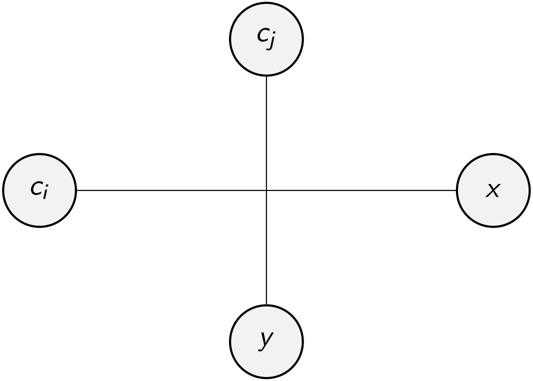
Given ϕ in 3CNF form we will

1. Draw the graph of ϕ as a grid (figure 1)
2. Note what crossings look like (figure 2)
3. Have a crossover gadget (figure 3).
4. Say what we add to the formula to get that crossover gadget.

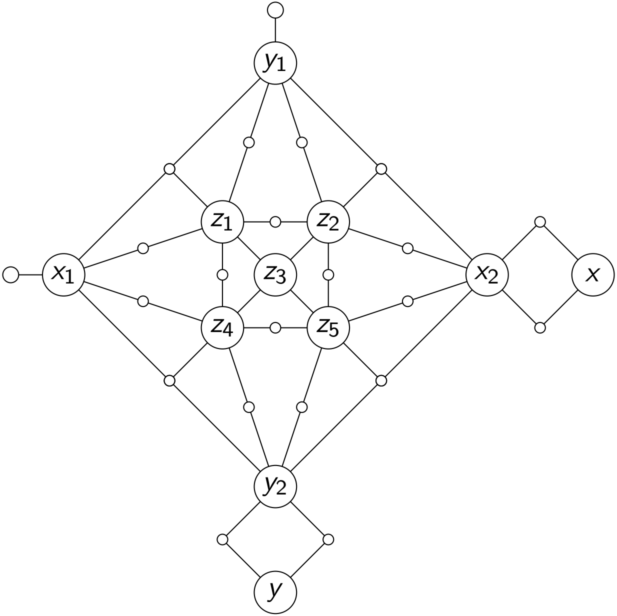
$$(v \vee \neg w \vee x) \wedge (v \vee \neg w \vee \neg y) \wedge (v \vee \neg x \vee z)$$



The Kinds of Crossings We Will Deal With



Crossover Gadget For Planar SAT



Add to the Boolean Fml

The crossover gadget does not tell us what to add to the formula since if C and x are connected then either x or $\overline{x}$ could be in C.

Add to the Boolean Fml

The crossover gadget does not tell us what to add to the formula since if C and x are connected then either x or $\overline{x}$ could be in C.

We add the following to the formula

$(\overline{x_2} \vee \overline{y_2} \vee z_5) \wedge (\overline{x_2} \vee \overline{z_5}) \wedge (y_2 \vee \overline{z_5})$ (which is $(x_2 \wedge y_2) \leftrightarrow z_5$)

Add to the Boolean Fml

The crossover gadget does not tell us what to add to the formula since if C and x are connected then either x or $\overline{x}$ could be in C. We add the following to the formula

$(\overline{x_2} \vee \overline{y_2} \vee z_5) \wedge (\overline{x_2} \vee \overline{z_5}) \wedge (y_2 \vee \overline{z_5})$ (which is $(x_2 \wedge y_2) \leftrightarrow z_5)$

$(\overline{x_2} \vee \overline{y_1} \vee z_2) \wedge (\overline{x_2} \vee \overline{z_2}) \wedge (\overline{y_1} \vee \overline{z_2})$ (which is $(x_2 \wedge \overline{y_2}) \leftrightarrow z_2)$

Add to the Boolean Fml

The crossover gadget does not tell us what to add to the formula since if C and x are connected then either x or $\overline{x}$ could be in C.
We add the following to the formula

$(\overline{x_2} \vee \overline{y_2} \vee z_5) \wedge (\overline{x_2} \vee \overline{z_5}) \wedge (y_2 \vee \overline{z_5})$ (which is $(x_2 \wedge y_2) \leftrightarrow z_5$)

$(\overline{x_2} \vee \overline{y_1} \vee z_2) \wedge (\overline{x_2} \vee \overline{z_2}) \wedge (\overline{y_1} \vee \overline{z_2})$ (which is $(x_2 \wedge \overline{y_2}) \leftrightarrow z_2$)

$(\overline{x_1} \vee \overline{y_1} \vee z_1) \wedge (\overline{x_1} \vee \overline{z_1}) \wedge (\overline{y_1} \vee \overline{z_1})$ (which is $(\overline{x_1} \wedge \overline{y_1}) \leftrightarrow z_1$)

Add to the Boolean Fml

The crossover gadget does not tell us what to add to the formula since if C and x are connected then either x or $\overline{x}$ could be in C. We add the following to the formula

$(\overline{x_2} \vee \overline{y_2} \vee z_5) \wedge (\overline{x_2} \vee \overline{z_5}) \wedge (y_2 \vee \overline{z_5})$ (which is $(x_2 \wedge y_2) \leftrightarrow z_5)$

$(\overline{x_2} \vee \overline{y_1} \vee z_2) \wedge (\overline{x_2} \vee \overline{z_2}) \wedge (\overline{y_1} \vee \overline{z_2})$ (which is $(x_2 \wedge \overline{y_2}) \leftrightarrow z_2)$

$(\overline{x_1} \vee \overline{y_1} \vee z_1) \wedge (\overline{x_1} \vee \overline{z_1}) \wedge (\overline{y_1} \vee \overline{z_1})$ (which is $(\overline{x_1} \wedge \overline{y_1}) \leftrightarrow z_1)$

$(\overline{x_1} \vee \overline{y_2} \vee z_4) \wedge (\overline{x_1} \vee \overline{z_4}) \wedge (\overline{y_2} \vee \overline{z_4})$ (which is $(\overline{x_1} \wedge y_2) \leftrightarrow z_4)$

Add to the Boolean Fml

The crossover gadget does not tell us what to add to the formula since if C and x are connected then either x or $\overline{x}$ could be in C.

We add the following to the formula

$(\overline{x_2} \vee \overline{y_2} \vee z_5) \wedge (\overline{x_2} \vee \overline{z_5}) \wedge (y_2 \vee \overline{z_5})$ (which is $(x_2 \wedge y_2) \leftrightarrow z_5$)

$(\overline{x_2} \vee \overline{y_1} \vee z_2) \wedge (\overline{x_2} \vee \overline{z_2}) \wedge (\overline{y_1} \vee \overline{z_2})$ (which is $(x_2 \wedge \overline{y_2}) \leftrightarrow z_2$)

$(\overline{x_1} \vee \overline{y_1} \vee z_1) \wedge (\overline{x_1} \vee \overline{z_1}) \wedge (\overline{y_1} \vee \overline{z_1})$ (which is $(\overline{x_1} \wedge \overline{y_1}) \leftrightarrow z_1$)

$(\overline{x_1} \vee \overline{y_2} \vee z_4) \wedge (\overline{x_1} \vee \overline{z_4}) \wedge (\overline{y_2} \vee \overline{z_4})$ (which is $(\overline{x_1} \wedge y_2) \leftrightarrow z_4$)

$(z_1 \vee z_2 \vee z_4 \vee z_5)$

Add to the Boolean Fml

The crossover gadget does not tell us what to add to the formula since if C and x are connected then either x or $\overline{x}$ could be in C. We add the following to the formula

$(\overline{x_2} \vee \overline{y_2} \vee z_5) \wedge (\overline{x_2} \vee \overline{z_5}) \wedge (y_2 \vee \overline{z_5})$ (which is $(x_2 \wedge y_2) \leftrightarrow z_5$)

$(\overline{x_2} \vee \overline{y_1} \vee z_2) \wedge (\overline{x_2} \vee \overline{z_2}) \wedge (\overline{y_1} \vee \overline{z_2})$ (which is $(x_2 \wedge \overline{y_2}) \leftrightarrow z_2$)

$(\overline{x_1} \vee \overline{y_1} \vee z_1) \wedge (\overline{x_1} \vee \overline{z_1}) \wedge (\overline{y_1} \vee \overline{z_1})$ (which is $(\overline{x_1} \wedge \overline{y_1}) \leftrightarrow z_1$)

$(\overline{x_1} \vee \overline{y_2} \vee z_4) \wedge (\overline{x_1} \vee \overline{z_4}) \wedge (\overline{y_2} \vee \overline{z_4})$ (which is $(\overline{x_1} \wedge y_2) \leftrightarrow z_4$)

$(z_1 \vee z_2 \vee z_4 \vee z_5)$

$(\overline{z_5} \vee \overline{z_2}) \wedge (\overline{z_2} \vee \overline{z_1}) \wedge (\overline{z_1} \vee z_5)$

Add to the Boolean Fml

The crossover gadget does not tell us what to add to the formula since if C and x are connected then either x or $\overline{x}$ could be in C. We add the following to the formula

$(\overline{x_2} \vee \overline{y_2} \vee z_5) \wedge (\overline{x_2} \vee \overline{z_5}) \wedge (y_2 \vee \overline{z_5})$ (which is $(x_2 \wedge y_2) \leftrightarrow z_5$)

$(\overline{x_2} \vee \overline{y_1} \vee z_2) \wedge (\overline{x_2} \vee \overline{z_2}) \wedge (\overline{y_1} \vee \overline{z_2})$ (which is $(x_2 \wedge \overline{y_2}) \leftrightarrow z_2$)

$(\overline{x_1} \vee \overline{y_1} \vee z_1) \wedge (\overline{x_1} \vee \overline{z_1}) \wedge (\overline{y_1} \vee \overline{z_1})$ (which is $(\overline{x_1} \wedge \overline{y_1}) \leftrightarrow z_1$)

$(\overline{x_1} \vee \overline{y_2} \vee z_4) \wedge (\overline{x_1} \vee \overline{z_4}) \wedge (\overline{y_2} \vee \overline{z_4})$ (which is $(\overline{x_1} \wedge y_2) \leftrightarrow z_4$)

$(z_1 \vee z_2 \vee z_4 \vee z_5)$

$(\overline{z_5} \vee \overline{z_2}) \wedge (\overline{z_2} \vee \overline{z_1}) \wedge (\overline{z_1} \vee z_5)$

$x_2 \vee \overline{x}) \wedge (x \vee \overline{x_2}) \wedge (y \vee \overline{y_2})$ (which is $x \leftrightarrow x_2$ and $y \leftrightarrow y_2$)

VC and Planar VC

Given ϕ we produce (G, k) such that

VC and Planar VC

Given ϕ we produce (G, k) such that

$$\phi \in \mathrm{3SAT} \text{ iff } (G, k) \in \mathrm{VC}$$

The same reduction will show

VC and Planar VC

Given ϕ we produce (G, k) such that

$$\phi \in \mathrm{3SAT} \text{ iff } (G, k) \in \mathrm{VC}$$

The same reduction will show

$$\phi \in \mathrm{PL\text{-}3SAT} \text{ iff } (G, k) \in \mathrm{PL\text{-}VC}$$

VC and Planar VC

Given ϕ we produce (G, k) such that

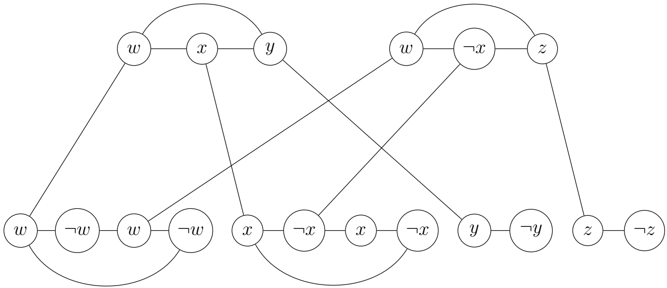
$$\phi \in \mathrm{3SAT} \text{ iff } (G, k) \in \mathrm{VC}$$

The same reduction will show

$$\phi \in \mathrm{PL\text{-}3SAT} \text{ iff } (G, k) \in \mathrm{PL\text{-}VC}$$

We give an example of the key gadget on the next slide.

$(w \vee x \vee y) \wedge (w \vee \neg x \vee z)$



General Construction for VC and Planar VC

Given $\phi = C_1 \wedge \cdots \wedge C_k$.

1) Clause C: have a triangle connecting the literals.

General Construction for VC and Planar VC

Given $\phi = C_1 \wedge \cdots \wedge C_k$.
1) Clause C: have a triangle connecting the literals.
2) Var x: have alt-cycle of $x, \neg x$ of length $2\times$(numb of x's).

General Construction for VC and Planar VC

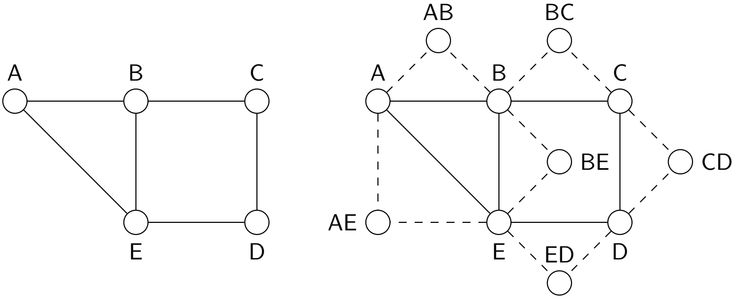
Given $\phi = C_1 \wedge \cdots \wedge C_k$.
1) Clause C: have a triangle connecting the literals.
2) Var x: have alt-cycle of $x, \neg x$ of length $2\times$(numb of x's).
3) If C has x draw line from one of the x to C. Diff for each C.

General Construction for VC and Planar VC

Given $\phi = C_1 \wedge \cdots \wedge C_k$.
1) Clause C: have a triangle connecting the literals.
2) Var x: have alt-cycle of $x, \neg x$ of length $2\times$(numb of x's).
3) If C has x draw line from one of the x to C. Diff for each C.
4) Seek a VC of size $5k$.

DOM and Planar DOM

Example of Gadget for DOM and Planar DOM



G =
Vertex Cover {B,C,E}

G' =
Dominating Set {B,E,CD}

Figure: Proof that Dominating Set is NP-Complete

General Construction for (Planar) DOM

Given VC instance (G, k) create G' as follows.
1) For every edge (a, b) create a new vertex ab that has an edge to a and b.

General Construction for (Planar) DOM

Given VC instance (G, k) create G' as follows.
1) For every edge (a, b) create a new vertex ab that has an edge to a and b.
2) G has a VC size $k \rightarrow G'$ as a DOM of size k: take the VC itself.

General Construction for (Planar) DOM

Given VC instance (G, k) create G' as follows.
1) For every edge (a, b) create a new vertex ab that has an edge to a and b.
2) G has a VC size $k \rightarrow G'$ as a DOM of size k: take the VC itself.
3) G' has a DOM of size $k \rightarrow G$ has a VC of size k: take the DOM set but if one of the vertices of form ab just take a.

Planar 1-in-3 SAT

Def Planar 1-in-3 SAT ($\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$): Given a Planar 3CNF formula ϕ, is there a satisfying assignment where every clause has exactly 1 literal T?

Planar 1-in-3 SAT

Def Planar 1-in-3 SAT ($\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$): Given a Planar 3CNF formula ϕ, is there a satisfying assignment where every clause has exactly 1 literal T?

Thm $\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$ is NPC.

Planar 1-in-3 SAT

Def Planar 1-in-3 SAT ($\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$): Given a Planar 3CNF formula ϕ, is there a satisfying assignment where every clause has exactly 1 literal T?

Thm $\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$ is NPC.

We show $\mathrm{PL\text{-}3SAT} \le \mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$.

Planar 1-in-3 SAT

Def Planar 1-in-3 SAT ($\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$): Given a Planar 3CNF formula ϕ, is there a satisfying assignment where every clause has exactly 1 literal T?

Thm $\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$ is NPC.

We show $\mathrm{PL\text{-}3SAT} \le \mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$.

Given ϕ, a planar 3CNF formula, we form ϕ':
Replace every clause $L_1 \vee L_2 \vee L_3$ in ϕ (the L_i are literals) with

$$(L_1 \vee a \vee b) \wedge (\overline{L_2} \vee a \vee c) \wedge (\overline{L_3} \vee b \vee d)$$

Planar 1-in-3 SAT

Def Planar 1-in-3 SAT ($\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$): Given a Planar 3CNF formula ϕ, is there a satisfying assignment where every clause has exactly 1 literal T?

Thm $\mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$ is NPC.

We show $\mathrm{PL\text{-}3SAT} \le \mathrm{PL\text{-}1\text{-}in\text{-}3SAT}$.

Given ϕ, a planar 3CNF formula, we form ϕ':
Replace every clause $L_1 \vee L_2 \vee L_3$ in ϕ (the L_i are literals) with

$$(L_1 \vee a \vee b) \wedge (\overline{L_2} \vee a \vee c) \wedge (\overline{L_3} \vee b \vee d)$$

Left to the reader to prove the ϕ' is planar and

$$\phi \in \mathrm{PL\text{-}3SAT} \text{ iff } \phi \in \mathrm{PL\text{-}1\text{-}in\text{-}3SAT}.$$

Exact Cover and Planar Exact Cover

Def $[n]$ is the set $\{1, \ldots, n\}$.
Def Exact Cover (X3C): Given $n \equiv 0 \pmod 3$ and a set $E_1, \ldots, E_m$ of 3-subsets of $[n]$ is does some set of $n/3$ of the E_i's cover $[n]$. Note that they cannot overlap.

Exact Cover and Planar Exact Cover

Def $[n]$ is the set $\{1, \ldots, n\}$.
Def Exact Cover ($\mathrm{X3C}$): Given $n \equiv 0 \pmod 3$ and a set $E_1, \ldots, E_m$ of 3-subsets of $[n]$ is does some set of $n/3$ of the E_i's cover $[n]$. Note that they cannot overlap.

Def We associate to an instance of $\mathrm{X3C}$ a bipartite graph:
$L = \{E_1, \ldots, E_m\}$, $R = [n]$,

Exact Cover and Planar Exact Cover

Def $[n]$ is the set $\{1, \ldots, n\}$.
Def Exact Cover ($\mathrm{X3C}$): Given $n \equiv 0 \pmod 3$ and a set $E_1, \ldots, E_m$ of 3-subsets of $[n]$ is does some set of $n/3$ of the E_i's cover $[n]$. Note that they cannot overlap.

Def We associate to an instance of $\mathrm{X3C}$ a bipartite graph:
$L = \{E_1, \ldots, E_m\}$, $R = [n]$,

But an edge between E_i and j if $j \in E_i$.

Exact Cover and Planar Exact Cover

Def $[n]$ is the set $\{1, \ldots, n\}$.
Def Exact Cover ($\mathrm{X3C}$): Given $n \equiv 0 \pmod 3$ and a set $E_1, \ldots, E_m$ of 3-subsets of $[n]$ is does some set of $n/3$ of the E_i's cover $[n]$. Note that they cannot overlap.

Def We associate to an instance of $\mathrm{X3C}$ a bipartite graph:
$L = \{E_1, \ldots, E_m\}$, $R = [n]$,

But an edge between E_i and j if $j \in E_i$.

Def Planar Exact Cover ($\mathrm{PL\text{-}X3C}$): Input is an instance of $\mathrm{X3C}$ where the graph is planar.

PL-X3C is NPC

We do 1-in-3-SAT $\le$ X3C. Modifying to get PL-1-in-3SAT $\le$ PL-X3C is not automatic but not that hard.

PL-X3C is NPC

We do 1-in-3-SAT $\le$ X3C. Modifying to get PL-1-in-3SAT $\le$ PL-X3C is not automatic but not that hard.

Warning It was impossible to get this onto a slide so the proof I am giving you is **missing some parts**, but the intuition is correct. For complete proof see the paper by Dyer and Frieze.
`https://www.math.cmu.edu/~af1p/Texfiles/3DM.pdf`

PL-X3C is NPC

We do 1-in-3-SAT $\le$ X3C. Modifying to get PL-1-in-3SAT $\le$ PL-X3C is not automatic but not that hard.

Warning It was impossible to get this onto a slide so the proof I am giving you is **missing some parts**, but the intuition is correct. For complete proof see the paper by Dyer and Frieze.
`https://www.math.cmu.edu/~af1p/Texfiles/3DM.pdf`

Given $C_1 \wedge \cdots \wedge C_m$ we create a instance of X3C.

PL-X3C is NPC

We do 1-in-3-SAT $\le$ X3C. Modifying to get
PL-1-in-3SAT $\le$ PL-X3C is not automatic but not that hard.

Warning It was impossible to get this onto a slide so the proof I am giving you is **missing some parts**, but the intuition is correct. For complete proof see the paper by Dyer and Frieze.
`https://www.math.cmu.edu/~af1p/Texfiles/3DM.pdf`

Given $C_1 \wedge \cdots \wedge C_m$ we create a instance of X3C.

Example is on next page.
Small circles ae elements.
Large circles labeled E are 3-sets. They contain what they have an edge to.

Example of Reduction

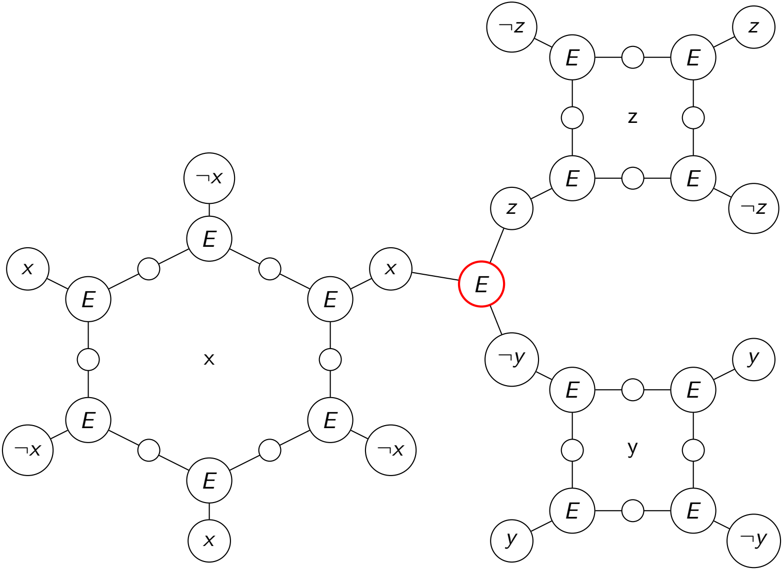


Figure: Gadget for PL-1-in-3SAT $\leq$ PL-X3C reduction.

Formal Reduction

Given $C_1 \wedge \cdots \wedge C_m$ we create a instance of $\mathrm{X3C}$.

Formal Reduction

Given $C_1 \wedge \cdots \wedge C_m$ we create a instance of $\mathrm{X3C}$.

1) For every variable x, if x appears m times then have a cycle of length $2m$ of E's and dummy elements. Then have $x, \neg x, x, \ldots, \neg x$ m of these. The E has in it the two dummies next to it and either x or $\neg x$.

Formal Reduction

Given $C_1 \wedge \cdots \wedge C_m$ we create a instance of $\mathrm{X3C}$.

1) For every variable x, if x appears m times then have a cycle of length $2m$ of E's and dummy elements. Then have $x, \neg x, x, \ldots, \neg x$ m of these. The E has in it the two dummies next to it and either x or $\neg x$.

2) For every clause $(L_1 \vee L_2 \vee L_3)$ there is a new set E which has in it an L_1, an $_2$, an L_3 from above. Each clause uses diff ones.

Formal Reduction

Given $C_1 \wedge \cdots \wedge C_m$ we create a instance of $\mathrm{X3C}$.

1) For every variable x, if x appears m times then have a cycle of length $2m$ of E's and dummy elements. Then have $x, \neg x, x, \ldots, \neg x$ m of these. The E has in it the two dummies next to it and either x or $\neg x$.

2) For every clause $(L_1 \vee L_2 \vee L_3)$ there is a new set E which has in it an L_1, an $_2$, an L_3 from above. Each clause uses diff ones.

3) Intuition: Assume there is a 1-in-3 SAT assignment. Let x be a var set T. Then all of the x's in the var-gadget will be covered by the clauses they appear in. So half of the E's in the var-gadget are used. (This is not quite right since we are assuming that x appears in exactly m clauses.)

Planar Bipartite DOM set (PL-bi-DOM)

Thm Planar bipartite DOM set is NPC.
We show that $\mathrm{PL\text{-}X3C} \leq \mathrm{PL\text{-}bi\text{-}DOM}$.

Given n and $E_1, \ldots, E_m$ we already have a planar bipartite graph.
For each i add edges (E_i, a_i) and (a_i, b_i).

Planar Bipartite DOM set (PL-bi-DOM)

Thm Planar bipartite DOM set is NPC.
We show that $\mathrm{PL\text{-}X3C} \le \mathrm{PL\text{-}bi\text{-}DOM}$.

Given n and $E_1, \ldots, E_m$ we already have a planar bipartite graph. For each i add edges (E_i, a_i) and (a_i, b_i).

Easy to show that there is a covering of size $n/3$ iff there is a DOM set of size $\frac{n}{3} + m$.

BILL AND NATHAN, RECORD LECTURE!!!!

BILL: EITHER GO TO GRIPNP PACKET OR STOP RECORDING LECTURE!!!