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

Exposition by William Gasarch—U of MD

$\mathrm{CLIQ} \le \mathrm{SAT}$. Why?

Bill Today we will prove $\mathrm{CLIQ} \le \mathrm{SAT}$.

CLIQ $\le$ SAT. Why?

Bill Today we will prove CLIQ $\le$ SAT.

Yaelle That's stupid! We know CLIQ $\le$ SAT by Cook-Levin.

$\mathrm{CLIQ} \le \mathrm{SAT}$. Why?

Bill Today we will prove $\mathrm{CLIQ} \le \mathrm{SAT}$.

Yaelle That's stupid! We know $\mathrm{CLIQ} \le \mathrm{SAT}$ by Cook-Levin.

Bill Write a program that will, given (G, k) produce ϕ such that

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

CLIQ $\le$ SAT. **Why?**

Bill Today we will prove $\mathrm{CLIQ} \le \mathrm{SAT}$.

Yaelle That's stupid! We know $\mathrm{CLIQ} \le \mathrm{SAT}$ by Cook-Levin.

Bill Write a program that will, given (G, k) produce ϕ such that

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

Yaelle Deal with Turing Machines? That's**insane!**

CLIQ $\le$ SAT. Why?

Bill Today we will prove $\mathrm{CLIQ} \le \mathrm{SAT}$.

Yaelle That's stupid! We know $\mathrm{CLIQ} \le \mathrm{SAT}$ by Cook-Levin.

Bill Write a program that will, given (G, k) produce ϕ such that

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

Yaelle Deal with Turing Machines? That's**insane!**

Bill Correct. I will show $\mathrm{CLIQ} \le \mathrm{SAT}$ in a **sane** way.

$\mathrm{CLIQ} \le \mathrm{SAT}$. Why?

Bill Today we will prove $\mathrm{CLIQ} \le \mathrm{SAT}$.

Yaelle That's stupid! We know $\mathrm{CLIQ} \le \mathrm{SAT}$ by Cook-Levin.

Bill Write a program that will, given (G, k) produce ϕ such that

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

Yaelle Deal with Turing Machines? That's**insane!**

Bill Correct. I will show $\mathrm{CLIQ} \le \mathrm{SAT}$ in a **sane** way.

Yaelle Why? Not practical since SAT is hard. Not theoretically interesting since we already know $\mathrm{CLIQ} \le \mathrm{SAT}$.

CLIQ $\le$ SAT. Why?

Bill Today we will prove $\text{CLIQ} \le \text{SAT}$.

Yaelle That's stupid! We know $\text{CLIQ} \le \text{SAT}$ by Cook-Levin.

Bill Write a program that will, given (G, k) produce ϕ such that

$$(G, k) \in \text{CLIQ} \text{ iff } \phi \in \text{SAT}$$

Yaelle Deal with Turing Machines? That's**insane!**

Bill Correct. I will show $\text{CLIQ} \le \text{SAT}$ in a **sane** way.

Yaelle Why? Not practical since SAT is hard. Not theoretically interesting since we already know $\text{CLIQ} \le \text{SAT}$.

Bill Because there are **awesome SAT Solvers!**

Old View, New View

Old View, New View

Old View I want to solve CLIQ. Since $\mathrm{SAT} \le \mathrm{CLIQ}$, CLIQ is probably hard. Darn!

Old View, New View

Old View I want to solve CLIQ. Since $\mathrm{SAT} \le \mathrm{CLIQ}$, CLIQ is probably hard. Darn!

New View I want to solve CLIQ. I know from Cook-Levin that $\mathrm{CLIQ} \le \mathrm{SAT}$.

Old View, New View

Old View I want to solve CLIQ. Since $\mathrm{SAT} \le \mathrm{CLIQ}$, CLIQ is probably hard. Darn!

New View I want to solve CLIQ. I know from Cook-Levin that $\mathrm{CLIQ} \le \mathrm{SAT}$.
That reduction is insane (hard and blow up).

Old View, New View

Old View I want to solve CLIQ. Since $\mathrm{SAT} \le \mathrm{CLIQ}$, CLIQ is probably hard. Darn!

New View I want to solve CLIQ. I know from Cook-Levin that $\mathrm{CLIQ} \le \mathrm{SAT}$.
That reduction is insane (hard and blow up).
If I can find a better reduction of $\mathrm{CLIQ} \le \mathrm{SAT}$ then to solve a CLIQ problem I can transform it to a SAT problem, and solve that.

Old View, New View

Old View I want to solve CLIQ. Since $\mathrm{SAT} \le \mathrm{CLIQ}$, CLIQ is probably hard. Darn!

New View I want to solve CLIQ. I know from Cook-Levin that $\mathrm{CLIQ} \le \mathrm{SAT}$.
That reduction is insane (hard and blow up).
If I can find a better reduction of $\mathrm{CLIQ} \le \mathrm{SAT}$ then to solve a CLIQ problem I can transform it to a SAT problem, and solve that.

Caveat This does not always work.

Old View, New View

Old View I want to solve CLIQ. Since $\mathrm{SAT} \le \mathrm{CLIQ}$, CLIQ is probably hard. Darn!

New View I want to solve CLIQ. I know from Cook-Levin that $\mathrm{CLIQ} \le \mathrm{SAT}$.
That reduction is insane (hard and blow up).
If I can find a better reduction of $\mathrm{CLIQ} \le \mathrm{SAT}$ then to solve a CLIQ problem I can transform it to a SAT problem, and solve that.

Caveat This does not always work.

1. SAT solvers are only good on some problems.

Old View, New View

Old View I want to solve CLIQ. Since $\mathrm{SAT} \le \mathrm{CLIQ}$, CLIQ is probably hard. Darn!

New View I want to solve CLIQ. I know from Cook-Levin that $\mathrm{CLIQ} \le \mathrm{SAT}$.
That reduction is insane (hard and blow up).
If I can find a better reduction of $\mathrm{CLIQ} \le \mathrm{SAT}$ then to solve a CLIQ problem I can transform it to a SAT problem, and solve that.

Caveat This does not always work.

1. SAT solvers are only good on some problems.
2. Getting the reductions to not blow up is not always possible.

How to View CLIQ

Does G have a clique of size k?

How to View CLIQ

Does G have a clique of size k?

We rephrase that:

How to View CLIQ

Does G have a clique of size k?

We rephrase that:

Let $G = (V, E)$.

How to View CLIQ

Does G have a clique of size k?

We rephrase that:

Let $G = (V, E)$.

G has a clique of size k is **equivalent** to:
There is a 1-1 function $\{1, \ldots, k\} \rightarrow V$ such that for all $1 \le a, b \le k$, $(f(a), f(b)) \in E$.

How to View CLIQ

Does G have a clique of size k?

We rephrase that:

Let $G = (V, E)$.

G has a clique of size k is **equivalent** to:
There is a 1-1 function $\{1, \ldots, k\} \rightarrow V$ such that for all $1 \le a, b \le k$, $(f(a), f(b)) \in E$.

I will go to the Zoom whiteboard and do an example, drawing with the mouse.

How to View CLIQ

Does G have a clique of size k?

We rephrase that:

Let $G = (V, E)$.

G has a clique of size k is **equivalent** to:
There is a 1-1 function $\{1, \ldots, k\} \rightarrow V$ such that for all $1 \leq a, b \leq k$, $(f(a), f(b)) \in E$.

I will go to the Zoom whiteboard and do an example, drawing with the mouse. Wish me luck.

CLIQ $\le$ SAT

We want to know:

Is there a 1-1 function $\{1, \ldots, k\} \rightarrow V$ such that for all $1 \le a, b \le k$, $(f(a), f(b)) \in E$.

CLIQ $\le$ SAT

We want to know:
Is there a 1-1 function $\{1, \ldots, k\} \rightarrow V$ such that for all $1 \le a, b \le k$, $(f(a), f(b)) \in E$.

We formulate this as a Boolean Formula.

CLIQ $\le$ SAT

We want to know:

Is there a 1-1 function $\{1,\ldots,k\} \rightarrow V$ such that for all $1 \le a,b \le k$, $(f(a), f(b)) \in E$.

We formulate this as a Boolean Formula.

For $1 \le i \le k$, $1 \le j \le n$, have Boolean Vars x_{ij}.

CLIQ $\le$ SAT

We want to know:

Is there a 1-1 function $\{1, \ldots, k\} \rightarrow V$ such that for all $1 \le a, b \le k$, $(f(a), f(b)) \in E$.

We formulate this as a Boolean Formula.

For $1 \le i \le k$, $1 \le j \le n$, have Boolean Vars x_{ij}.

Intent

$$x_{ij} = \begin{cases} T & \text{if numb } i \text{ maps to vertex } j \\ F & \text{if numb } i \text{ does not maps to vertex } j \end{cases} \qquad (1)$$

Formula: x_{ij} Represent a 1-1 Function

The formula is in diff parts to guarantee diff things.

Formula: x_{ij} Represent a 1-1 Function

The formula is in diff parts to guarantee diff things.

Every *i* maps to at least one *j*

For $1 \leq i \leq k$

$$x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$$

Formula: x_{ij} Represent a 1-1 Function

The formula is in diff parts to guarantee diff things.

Every i maps to at least one j

For $1 \le i \le k$

$$x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$$

Every i maps to at at most one j

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$

$$\neg(x_{ij_1} \wedge x_{ij_2})$$

Formula: x_{ij} Represent a 1-1 Function

The formula is in diff parts to guarantee diff things.

Every i maps to at least one j

For $1 \le i \le k$

$$x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$$

Every i maps to at at most one j

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$

$$\neg(x_{ij_1} \wedge x_{ij_2})$$

The mapping is 1-1

For $1 \le i_1 < i_2 \le k$, for $1 \le j \le n$

$$\neg(x_{i_1,j} \wedge x_{i_2,j})$$

Formula: x_{ij} Represent a 1-1 Function

The formula is in diff parts to guarantee diff things.

Every i maps to at least one j
For $1 \le i \le k$

$$x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$$

Every i maps to at at most one j
For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$

$$\neg(x_{ij_1} \wedge x_{ij_2})$$

The mapping is 1-1
For $1 \le i_1 < i_2 \le k$, for $1 \le j \le n$

$$\neg(x_{i_1,j} \wedge x_{i_2,j})$$

Note So far all we've used about G is that it has n vertices.

Formula: The Edges are Preserved

We need that if i_1 maps to j_1 and i_2 maps to j_2 then $(j_1, j_2) \in E$.

Formula: The Edges are Preserved

We need that if i_1 maps to j_1 and i_2 maps to j_2 then $(j_1, j_2) \in E$.

For every $1 \leq i_1 < i_2 \leq k$

$$\bigvee_{(j_1,j_2) \in E} x_{i_1 j_1} \wedge x_{i_2 j_2}.$$

How Big is the Formula

We state the parts of the formula and how long they are.

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \le i \le k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \le i \le k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$ $\neg(x_{ij_1} \wedge x_{ij_2})$. $O(kn^2)$

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \le i \le k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$ $\neg(x_{ij_1} \wedge x_{ij_2})$. $O(kn^2)$

For $1 \le i_1 < i_2 \le k$, for $1 \le j \le n$ $\neg(x_{i_1,j} \wedge x_{i_2,j})$. $O(k^2 n)$

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \le i \le k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$ $\neg(x_{ij_1} \wedge x_{ij_2})$. $O(kn^2)$

For $1 \le i_1 < i_2 \le k$, for $1 \le j \le n$ $\neg(x_{i_1,j} \wedge x_{i_2,j})$. $O(k^2n)$

For every $1 \le i_1 < i_2 \le k$, $\bigvee_{(j_1,j_2)\in E} x_{i_1j_1} \wedge x_{i_2j_2}$. $O(k^2|E|)$

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \leq i \leq k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

For $1 \leq i \leq k$, for $1 \leq j_1 < j_2 \leq n$ $\neg(x_{ij_1} \wedge x_{ij_2})$. $O(kn^2)$

For $1 \leq i_1 < i_2 \leq k$, for $1 \leq j \leq n$ $\neg(x_{i_1,j} \wedge x_{i_2,j})$. $O(k^2n)$

For every $1 \leq i_1 < i_2 \leq k$, $\bigvee_{(j_1,j_2)\in E} x_{i_1j_1} \wedge x_{i_2j_2}$. $O(k^2|E|)$

- The formula is of size $O(kn^2) + O(k^2n) + O(k^2|E|)$.

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \le i \le k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$ $\neg(x_{ij_1} \wedge x_{ij_2})$. $O(kn^2)$

For $1 \le i_1 < i_2 \le k$, for $1 \le j \le n$ $\neg(x_{i_1,j} \wedge x_{i_2,j})$. $O(k^2n)$

For every $1 \le i_1 < i_2 \le k$, $\bigvee_{(j_1,j_2)\in E} x_{i_1j_1} \wedge x_{i_2j_2}$. $O(k^2|E|)$

- The formula is of size $O(kn^2) + O(k^2n) + O(k^2|E|)$.
- The construction is easy to do. Yaelle **could** code this up.

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \le i \le k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$ $\neg(x_{ij_1} \wedge x_{ij_2})$. $O(kn^2)$

For $1 \le i_1 < i_2 \le k$, for $1 \le j \le n$ $\neg(x_{i_1,j} \wedge x_{i_2,j})$. $O(k^2 n)$

For every $1 \le i_1 < i_2 \le k$, $\bigvee_{(j_1,j_2)\in E} x_{i_1 j_1} \wedge x_{i_2 j_2}$. $O(k^2|E|)$

- The formula is of size $O(kn^2) + O(k^2 n) + O(k^2|E|)$.
- The construction is easy to do. Yaelle **could** code this up.
- The constants are small.

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \le i \le k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$ $\neg(x_{ij_1} \wedge x_{ij_2})$. $O(kn^2)$

For $1 \le i_1 < i_2 \le k$, for $1 \le j \le n$ $\neg(x_{i_1,j} \wedge x_{i_2,j})$. $O(k^2 n)$

For every $1 \le i_1 < i_2 \le k$, $\bigvee_{(j_1,j_2)\in E} x_{i_1 j_1} \wedge x_{i_2 j_2}$. $O(k^2|E|)$

- The formula is of size $O(kn^2) + O(k^2 n) + O(k^2|E|)$.
- The construction is easy to do. Yaelle **could** code this up.
- The constants are small.
- Usually $k \ll n$ so the real issue is the n^2 and the $|E|$.

How Big is the Formula

We state the parts of the formula and how long they are.

For $1 \le i \le k$: $x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$. $O(kn)$.

For $1 \le i \le k$, for $1 \le j_1 < j_2 \le n$ $\neg(x_{ij_1} \wedge x_{ij_2})$. $O(kn^2)$

For $1 \le i_1 < i_2 \le k$, for $1 \le j \le n$ $\neg(x_{i_1,j} \wedge x_{i_2,j})$. $O(k^2 n)$

For every $1 \le i_1 < i_2 \le k$, $\bigvee_{(j_1,j_2) \in E} x_{i_1 j_1} \wedge x_{i_2 j_2}$. $O(k^2|E|)$

- The formula is of size $O(kn^2) + O(k^2 n) + O(k^2|E|)$.
- The construction is easy to do. Yaelle **could** code this up.
- The constants are small.
- Usually $k \ll n$ so the real issue is the n^2 and the $|E|$.
- Upshot: probably really good on sparse graphs.