On P vs. NP

If 50 million people believe a foolish thing,
it's still a foolish thing.
— George Bernard Shaw (1856–1950)

Exponential Circuit Complexity for NP-Complete Problems

- We shall prove exponential lower bounds for NP-complete problems using *monotone* circuits.
  - Monotone circuits are circuits without $\neg$ gates.[a]
- Note that this result does *not* settle the P vs. NP problem.

---

[a]Recall p. 331.

The Power of Monotone Circuits

- Monotone circuits can only compute monotone boolean functions.
- They are powerful enough to solve a P-complete problem: MONOTONE CIRCUIT VALUE (p. 332).
- There are NP-complete problems that are not monotone; they cannot be computed by monotone circuits at all.
- There are NP-complete problems that are monotone; they can be computed by monotone circuits.
  - HAMILTONIAN PATH and CLIQUE.

$\text{CLIQUE}_{n,k}$

- $\text{CLIQUE}_{n,k}$ is the boolean function deciding whether a graph $G = (V, E)$ with n nodes has a clique of size k.
- The input gates are the $\binom{n}{2}$ entries of the adjacency matrix of G.
  - Gate g_{ij} is set to true if the associated undirected edge $\{ i, j \}$ exists.
- $\text{CLIQUE}_{n,k}$ is a monotone function.
- Thus it can be computed by a monotone circuit.
- Of course, this does not rule out that *non*monotone circuits for $\text{CLIQUE}_{n,k}$ may use *fewer* gates.

Crude Circuits

- One possible circuit for $\text{CLIQUE}_{n,k}$ does the following.
  1. For each $S \subseteq V$ with $|S| = k$, there is a circuit with $O(k^2)$ $\wedge$-gates testing whether S forms a clique.
  2. We then take an OR of the outcomes of all the $\binom{n}{k}$ subsets $S_1, S_2, \ldots, S_{\binom{n}{k}}$.
- This is a monotone circuit with $O(k^2 \binom{n}{k})$ gates, which is exponentially large unless k or $n - k$ is a constant.
- A **crude circuit** $\text{CC}(X_1, X_2, \ldots, X_m)$ tests if there is an $X_i \subseteq V$ that forms a clique.[a]
  - The above-mentioned circuit is $\text{CC}(S_1, S_2, \ldots, S_{\binom{n}{k}})$.

[a] Consider the empty set a clique.

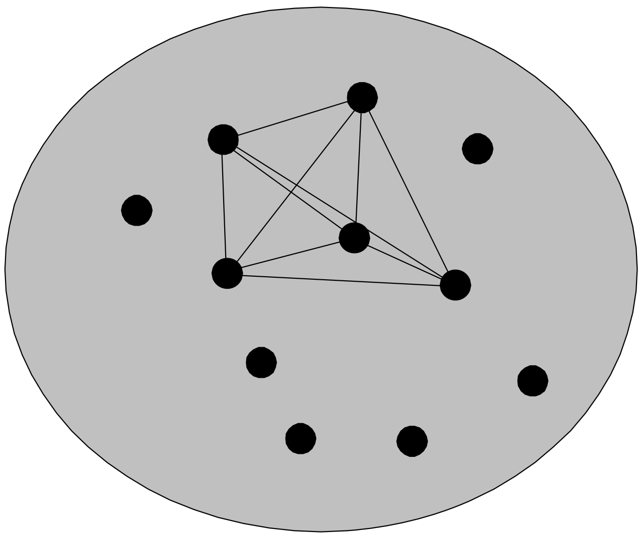
The Proof: Positive Examples

- Analysis will be applied to only the following **positive examples** and **negative examples** as input graphs.
- A positive example is a graph that has $\binom{k}{2}$ edges connecting k nodes in all possible ways.
- There are $\binom{n}{k}$ such graphs.
- $\textsc{clique}_{n,k}$ should output true on them.

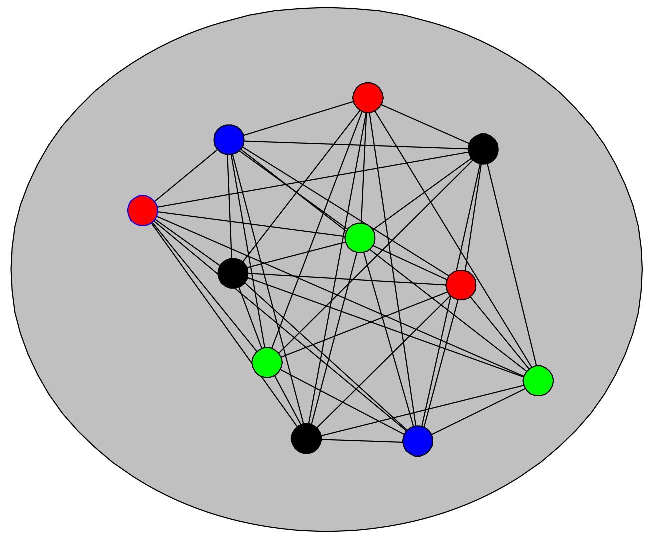
The Proof: Negative Examples

- Color the nodes with $k-1$ different colors and join by an edge any two nodes that are colored differently.
- There are $(k-1)^n$ such graphs.
- $\textsc{clique}_{n,k}$ should output false on them.
  - Each set of k nodes must have 2 identically colored nodes; hence there is no edge between them.

Positive and Negative Examples with $k = 5$



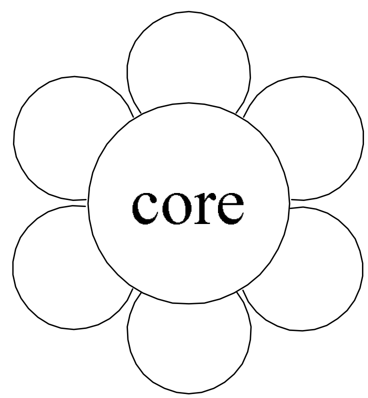
A positive example



A negative example

Sunflowers

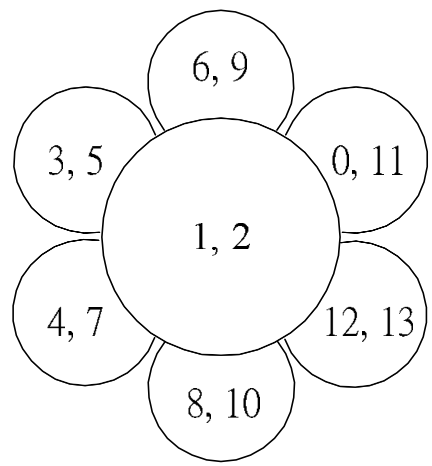
- Fix $p \in \mathbb{Z}^+$ and $\ell \in \mathbb{Z}^+$.
- A **sunflower** is a family of p sets $\{ P_1, P_2, \ldots, P_p \}$, called **petals**, each of cardinality at most ℓ.
- Furthermore, all pairs of sets in the family must have the same intersection (called the **core**[a] of the sunflower).



[a] A core can be an empty set.

A Sample Sunflower

$$\{\{1,2,3,5\},\{1,2,6,9\},\{0,1,2,11\},$$
$$\{1,2,12,13\},\{1,2,8,10\},\{1,2,4,7\}\}.$$



The Erdős-Rado Lemma

Lemma 86 *Let $\mathcal{Z}$ be a family of more than $M \stackrel{\Delta}{=} (p-1)^{\ell}\ell!$ nonempty sets, each of cardinality ℓ or less. Then $\mathcal{Z}$ must contain a sunflower (with p petals).*

- Induction on ℓ.
- For $\ell = 1$, p different singletons form a sunflower (with an empty core).
- Suppose $\ell > 1$.
- Consider a *maximal* subset $\mathcal{D} \subseteq \mathcal{Z}$ of *disjoint* sets.
  - Every set in $\mathcal{Z} - \mathcal{D}$ intersects some set in $\mathcal{D}$.

The Proof of the Erdős-Rado Lemma (continued)

For example,

$$\begin{aligned}
\mathcal{Z} &= \{\{1,2,3,5\},\{1,3,6,9\},\{0,4,8,11\},\\
&\quad \{4,5,6,7\},\{5,8,9,10\},\{6,7,9,11\}\},\\
\mathcal{D} &= \{\{1,2,3,5\},\{0,4,8,11\}\}.
\end{aligned}$$

The Proof of the Erdős-Rado Lemma (continued)

- Suppose $\mathcal{D}$ contains at least p sets.
  - $\mathcal{D}$ constitutes a sunflower with an empty core.
- Suppose $\mathcal{D}$ contains fewer than p sets.
  - Let C be the union of all sets in $\mathcal{D}$.
  - $|\, C \,| \leq (p-1)\ell$.
  - C intersects every set in $\mathcal{Z}$ by $\mathcal{D}$'s maximality.
  - There is a $d \in C$ that intersects more than $\frac{M}{(p-1)\ell} = (p-1)^{\ell-1}(\ell-1)!$ sets in $\mathcal{Z}$.
  - Consider $\mathcal{Z}' = \{\, Z - \{\, d \,\} : Z \in \mathcal{Z}, d \in Z \,\}$.

The Proof of the Erdős-Rado Lemma (concluded)

- (continued)
  - $\mathcal{Z}'$ has more than $M' \stackrel{\Delta}{=} (p-1)^{\ell-1}(\ell-1)!$ sets.
  - M' is just M with ℓ replaced with $\ell - 1$.
  - $\mathcal{Z}'$ contains a sunflower by induction, say

$$\{ P_1, P_2, \ldots, P_p \}.$$

  - Now,

$$\{ P_1 \cup \{ d \}, P_2 \cup \{ d \}, \ldots, P_p \cup \{ d \} \}$$

    is a sunflower in $\mathcal{Z}$.

Comments on the Erdős-Rado Lemma

- A family of more than M sets must contain a sunflower.
- **Plucking** a sunflower means replacing the sets in the sunflower by its core.
- By *repeatedly* finding a sunflower and plucking it, we can reduce a family with more than M sets to a family with at most M sets.
- If $\mathcal{Z}$ is a family of sets, the above result is denoted by pluck($\mathcal{Z}$).
- pluck($\mathcal{Z}$) is not unique.[a]

[a] It depends on the sequence of sunflowers one plucks. Fortunately, this issue is not material to the proof.

An Example of Plucking

- Recall the sunflower on p. 814:

$$\begin{aligned}\mathcal{Z} &= \{\{1,2,3,5\},\{1,2,6,9\},\{0,1,2,11\},\\ &\quad \{1,2,12,13\},\{1,2,8,10\},\{1,2,4,7\}\}\end{aligned}$$

- Then

$$\text{pluck}(\mathcal{Z}) = \{\{1,2\}\}.$$

Razborov's Theorem

Theorem 87 (Razborov, 1985) *There is a constant c such that for large enough n, all monotone circuits for* $\text{CLIQUE}_{n,k}$ *with $k = n^{1/4}$ have size at least $n^{cn^{1/8}}$.*

- We shall approximate any monotone circuit for $\text{CLIQUE}_{n,k}$ by a restricted kind of crude circuit.
- The approximation will proceed in steps: one step for each gate of the monotone circuit.
- Each step introduces few errors (false positives and false negatives).
- Yet, the final crude circuit has exponentially many errors.

The Proof

- Fix $k = n^{1/4}$.
- Fix $\ell = n^{1/8}$.
- Note that[a]
$$2\binom{\ell}{2} \leq k - 1. \tag{24}$$
- p will be fixed later to be $n^{1/8} \log n$.
- Fix $M = (p-1)^{\ell} \ell!$.
  - Recall the Erdős-Rado lemma (p. 815).

[a] Corrected by Mr. Moustapha Bande (`D98922042`) on January 5, 2010.

The Proof (continued)

- Each crude circuit used in the approximation process is of the form $\mathrm{CC}(X_1, X_2, \ldots, X_m)$, where:
  - $X_i \subseteq V$.
  - $|\, X_i \,| \leq \ell$.
  - $m \leq M$.
- It answers true if and only if at least one X_i is a clique.
- We shall show how to approximate any monotone circuit for $\textsc{clique}_{n,k}$ by such a crude circuit, inductively.
- The induction basis is straightforward:
  - Input gate g_{ij} is the crude circuit $\mathrm{CC}(\{i, j\})$.

The Proof (continued)

- A monotone circuit is the OR or AND of two subcircuits.
- We will build approximators of the overall circuit from the approximators of the two subcircuits.
  - Start with two crude circuits $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$.
  - $\mathcal{X}$ and $\mathcal{Y}$ are two families of at most M sets of nodes, each set containing at most ℓ nodes.
  - We will construct the approximate OR and the approximate AND of these subcircuits.
  - Then show both approximations introduce few errors.

The Proof: OR

- CC($\mathcal{X} \cup \mathcal{Y}$) is *equivalent to* the OR of CC($\mathcal{X}$) and CC($\mathcal{Y}$).
  - For any node set $\mathcal{C}$, $\mathcal{C} \in \mathcal{X} \cup \mathcal{Y}$ if and only if $\mathcal{C} \in \mathcal{X}$ or $\mathcal{C} \in \mathcal{Y}$.
  - Hence $\mathcal{X} \cup \mathcal{Y}$ contains a clique if and only if $\mathcal{X}$ or $\mathcal{Y}$ contains a clique.
- Problem with CC($\mathcal{X} \cup \mathcal{Y}$) occurs when $|\, \mathcal{X} \cup \mathcal{Y} \,| > M$.
- Such violations are eliminated by using

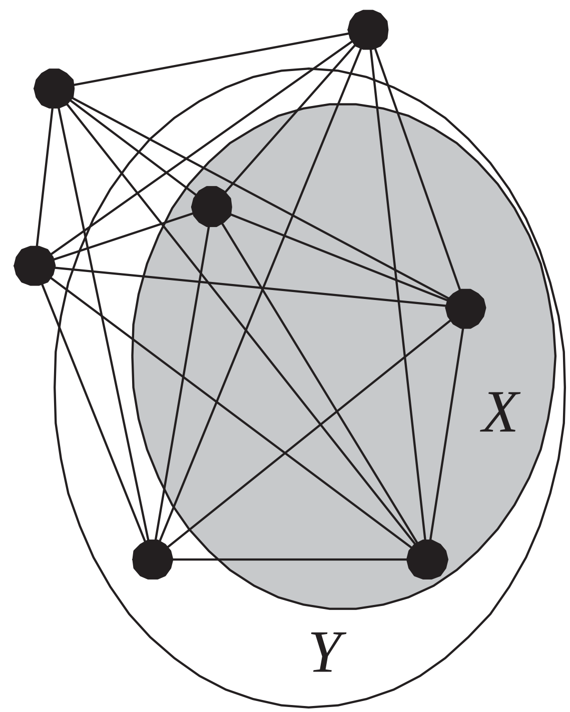
$$\text{CC}(\text{pluck}(\mathcal{X} \cup \mathcal{Y}))$$

as the final approximate OR of CC($\mathcal{X}$) and CC($\mathcal{Y}$).

The Proof: OR (continued)

- If $\mathrm{CC}(\mathcal{Z})$ is true, then $\mathrm{CC}(\mathrm{pluck}(\mathcal{Z}))$ must be true.
  - Each plucking replaces sets by their *common* core.
  - Let $Y \in \mathcal{Z}$ be a clique.
  - But a subset of Y must also be a clique.
  - So $\mathrm{pluck}(\mathcal{Z})$ must contain a clique.

The Proof: OR (continued)



The Proof: OR (concluded)

- CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces* a **false positive** if a negative example makes both CC($\mathcal{X}$) and CC($\mathcal{Y}$) return false but makes CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) return true.
- CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces* a **false negative** if a positive example makes either CC($\mathcal{X}$) or CC($\mathcal{Y}$) return true but makes CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) return false.
- We next count the number of false positives and false negatives introduced[a] by CC(pluck($\mathcal{X} \cup \mathcal{Y}$)).
- Let us work on false negatives for OR first.

[a]Compared with CC($\mathcal{X} \cup \mathcal{Y}$) of course.

The Number of False Negatives[a]

Lemma 88 CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces no false negatives.*

- Each plucking replaces sets in a crude circuit by their common subset.
- This makes the test for cliqueness less stringent.[b]

---

[a] Recall that CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) introduces a false negative if a positive example makes either CC($\mathcal{X}$) or CC($\mathcal{Y}$) return true but makes CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) return false.

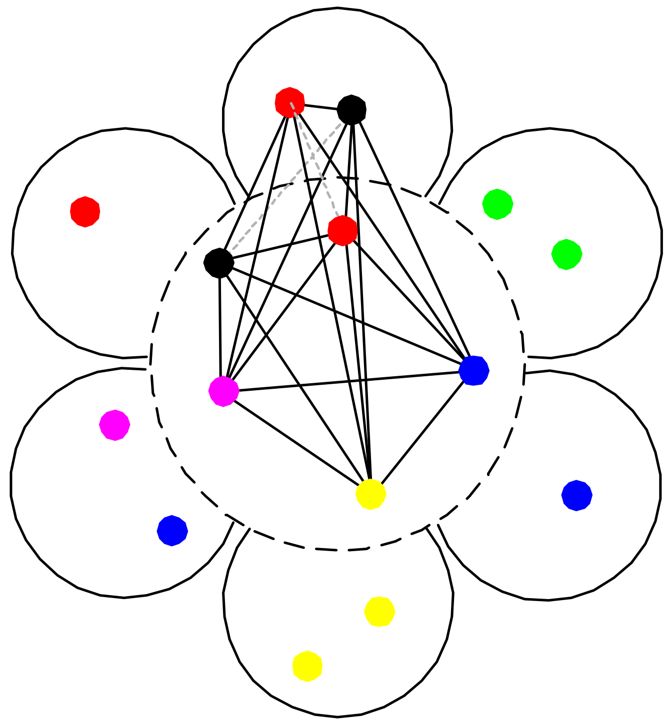
[b] The new crude circuit is at least as positive as the original one (p. 826).

The Number of False Positives

Lemma 89 CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces at most* $\frac{2M}{p-1} 2^{-p}(k-1)^n$ *false positives.*

- Each plucking operation replaces the sunflower $\{ Z_1, Z_2, \ldots, Z_p \}$ with its common core Z.
- A false positive is *necessarily* a coloring such that:
  - There is a pair of identically colored nodes in *each* petal Z_i (and so CC($Z_1, Z_2, \ldots, Z_p$) returns false).
  - But the core contains distinctly colored nodes (thus forming a clique).
  - This implies at least one node from each identical-color pair was plucked away.

Proof of Lemma 89 (continued)



Proof of Lemma 89 (continued)

- We now count the number of such colorings.
- Color nodes in V at random with $k - 1$ colors.
- Let $R(X)$ denote the event that there are repeated colors in set X.

Proof of Lemma 89 (continued)

- Now

$$
\begin{aligned}
& \text{prob}[\, R(Z_1) \wedge \cdots \wedge R(Z_p) \wedge \neg R(Z)\,] & (25)\\
\leq\; & \text{prob}[\, R(Z_1) \wedge \cdots \wedge R(Z_p) \,|\, \neg R(Z)\,] \\
=\; & \prod_{i=1}^{p} \text{prob}[\, R(Z_i) \,|\, \neg R(Z)\,] \\
\leq\; & \prod_{i=1}^{p} \text{prob}[\, R(Z_i)\,]. & (26)
\end{aligned}
$$

  – Equality holds because $R(Z_i)$ are independent given $\neg R(Z)$ as core Z contains their *only common* nodes.
  – Last inequality holds as the likelihood of repetitions in Z_i decreases given no repetitions in a subset, Z.

Proof of Lemma 89 (continued)

- Consider two nodes in Z_i.
- The probability that they have identical color is
$$\frac{1}{k-1}.$$
- Now
$$\text{prob}[\, R(Z_i)\,] \leq \frac{\binom{|Z_i|}{2}}{k-1} \leq \frac{\binom{\ell}{2}}{k-1} \leq \frac{1}{2} \tag{27}$$
by inequality (24) on p. 822.
- So the probability[a] that a random coloring yields a *new* false positive is at most 2^{-p} by inequality (26) on p. 833.

[a]Proportion, if you so prefer.

Proof of Lemma 89 (continued)

- As there are $(k-1)^n$ different colorings, *each* plucking introduces at most $2^{-p}(k-1)^n$ false positives.
- Recall that $|\,\mathcal{X} \cup \mathcal{Y}\,| \leq 2M$.
- When the procedure pluck($\mathcal{X} \cup \mathcal{Y}$) ends, the set system contains $\leq M$ sets.

Proof of Lemma 89 (concluded)

- Each plucking reduces the number of sets by $p - 1$.
- Hence at most $2M/(p-1)$ pluckings occur in pluck$(\mathcal{X} \cup \mathcal{Y})$.
- At most
$$\frac{2M}{p-1}\, 2^{-p}(k-1)^n$$
false positives are introduced.[a]

---

[a] Note that the numbers of errors are added not multiplied. Recall that we count how many *new* errors are introduced by each approximation step. Contributed by Mr. Ren-Shuo Liu (`D98922016`) on January 5, 2010.

The Proof: AND

- The approximate AND of crude circuits $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$ is

$$\text{CC}(\text{pluck}(\{\, X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, \mid X_i \cup Y_j \mid \leq \ell \,\})).$$

- We need to count the number of errors this approximate AND introduces on the positive and negative examples.

The Proof: AND (continued)

- The approximate AND *introduces* a **false positive** if a negative example makes either $\text{CC}(\mathcal{X})$ or $\text{CC}(\mathcal{Y})$ return false but makes the approximate AND return true.
- The approximate AND *introduces* a **false negative** if a positive example makes both $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$ return true but makes the approximate AND return false.
- As we count only new errors, we ignore scenarios where the AND of $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$ is already wrong.

The Proof: AND (continued)

- $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ introduces no false positives over our negative examples.[a]
  - Suppose $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ returns true.
  - Then some $X_i \cup Y_j$ is a clique.
  - Thus $X_i \in \mathcal{X}$ and $Y_j \in \mathcal{Y}$ are cliques, making both $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$ return true.
  - So $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ introduces no false positives.

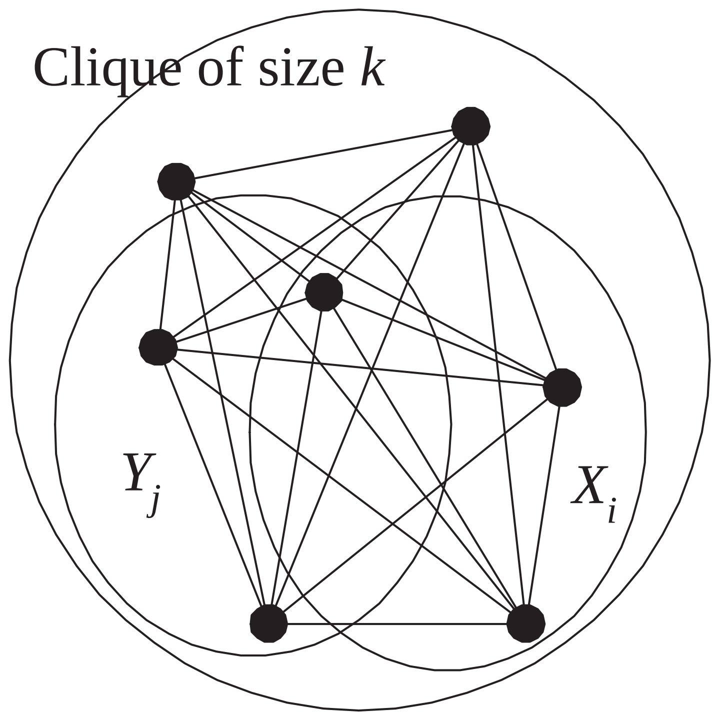
---

[a] Unlike the OR case on p. 825, we are not claiming that $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ is *equivalent to* the AND of $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$. Equivalence is more than we need here.

The Proof: AND (concluded)

- $\mathrm{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ introduces no false negatives over our positive examples.
  - Suppose *both* $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$ accept a positive example with a clique $\mathcal{C}$ of size k.
  - This clique $\mathcal{C}$ must contain an $X_i \in \mathcal{X}$ and a $Y_j \in \mathcal{Y}$.
  - As this clique $\mathcal{C}$ also contains $X_i \cup Y_j$ (see next page), the new circuit returns true.
  - $\mathrm{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ introduces no false negatives.
- We next bound the number of false positives and false negatives introduced[a] by the approximate AND.

[a] Compared with $\mathrm{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ of course.


Clique of size k
Y_j
X_i

The Number of False Positives

Lemma 90 *The approximate* AND *introduces at most* $M^2 2^{-p}(k-1)^n$ *false positives.*

- We prove this claim in stages.
- We knew $\mathrm{CC}(\{\, X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \,\})$ introduces no false positives.[a]
- $\mathrm{CC}(\{\, X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |\, X_i \cup Y_j \,| \leq \ell \,\})$ introduces no *additional* false positives because we are testing potentially *fewer* sets for cliqueness.

---

[a] Recall p. 839.

Proof of Lemma 90 (concluded)

- $|\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell \}| \leq M^2$.
- Each plucking reduces the number of sets by $p - 1$.
- So pluck($\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell \}$) involves $\leq M^2/(p-1)$ pluckings.
- Each plucking introduces at most $2^{-p}(k-1)^n$ false positives by the proof of Lemma 89 (p. 830).
- The desired upper bound is

$$[\, M^2/(p-1) \,]\, 2^{-p}(k-1)^n \leq M^2 2^{-p}(k-1)^n.$$

The Number of False Negatives

Lemma 91 *The approximate* AND *introduces at most* $M^2\binom{n-\ell-1}{k-\ell-1}$ *false negatives.*

- We again prove this claim in stages.
- We knew $\mathrm{CC}(\{\, X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \,\})$ introduces no false negatives.[a]

---

[a] Recall p. 839.

Proof of Lemma 91 (continued)

- $\text{CC}(\{\, X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |\, X_i \cup Y_j \,| \leq \ell \,\})$ introduces $\leq M^2 \binom{n-\ell-1}{k-\ell-1}$ false negatives.
  - Deletion of set $Z \stackrel{\Delta}{=} X_i \cup Y_j$ larger than ℓ introduces false negatives *only if* Z is part of a clique.
  - There are $\binom{n-|\,Z\,|}{k-|\,Z\,|}$ such cliques.
    * It is the number of positive examples whose clique contains Z.
  - $\binom{n-|\,Z\,|}{k-|\,Z\,|} \leq \binom{n-\ell-1}{k-\ell-1}$ as $|\,Z\,| > \ell$.
  - There are at most M^2 such Zs.

Proof of Lemma 91 (concluded)

- Plucking introduces no false negatives.
  - Recall that if $\mathrm{CC}(\mathcal{Z})$ is true, then $\mathrm{CC}(\mathrm{pluck}(\mathcal{Z}))$ must be true.[a]

---

[a]Recall p. 826.

Two Summarizing Lemmas

From Lemmas 89 (p. 830) and 90 (p. 842), we have:

Lemma 92 *Each approximation step introduces at most* $M^2 2^{-p}(k-1)^n$ *false positives.*

From Lemmas 88 (p. 829) and 91 (p. 844), we have:

Lemma 93 *Each approximation step introduces at most* $M^2\binom{n-\ell-1}{k-\ell-1}$ *false negatives.*

The Proof (continued)

- So each approximation step introduces "few" false positives and false negatives.
- We next show that the resulting crude circuit has "a lot" of false positives or false negatives.

The Final Crude Circuit

Lemma 94 *Every final crude circuit is:*

1. *Identically false—thus wrong on all positive examples.*
2. *Or outputs true on at least half of the negative examples.*

- Suppose it is not identically false.
- Then it accepts at least those graphs that have a clique on some set X of nodes, with

$$|\,X\,| \leq \ell = n^{1/8} < n^{1/4} = k.$$

Proof of Lemma 94 (concluded)

- Inequality (27) (p. 834) says that at least half of the colorings assign different colors to nodes in X.
- So at least half of the colorings — thus negative examples — have a clique in X and are accepted.

The Proof (continued)

- Recall the constants on p. 822:

$$
\begin{aligned}
k &\stackrel{\Delta}{=} n^{1/4}, \\
\ell &\stackrel{\Delta}{=} n^{1/8}, \\
p &\stackrel{\Delta}{=} n^{1/8}\log n, \\
M &\stackrel{\Delta}{=} (p-1)^{\ell}\ell! < n^{(1/3)n^{1/8}} \quad \text{for large } n.
\end{aligned}
$$

The Proof (continued)

- Suppose the final crude circuit is identically false.
  - By Lemma 93 (p. 847), each approximation step introduces at most $M^2\binom{n-\ell-1}{k-\ell-1}$ false negatives.
  - There are $\binom{n}{k}$ positive examples.
  - The original monotone circuit for $\text{CLIQUE}_{n,k}$ has at least

$$\frac{\binom{n}{k}}{M^2\binom{n-\ell-1}{k-\ell-1}} \geq \frac{1}{M^2}\left(\frac{n-\ell}{k}\right)^{\ell} \geq n^{(1/12)n^{1/8}}$$

    gates for large n.

The Proof (concluded)

- Suppose the final crude circuit is not identically false.
  - Lemma 94 (p. 849) says that there are at least $(k-1)^n/2$ false positives.
  - By Lemma 92 (p. 847), each approximation step introduces at most $M^2 2^{-p}(k-1)^n$ false positives
  - The original monotone circuit for $\text{CLIQUE}_{n,k}$ has at least

$$\frac{(k-1)^n/2}{M^2 2^{-p}(k-1)^n} = \frac{2^{p-1}}{M^2} \geq n^{(1/3)n^{1/8}}$$

gates.

Alexander Razborov (1963–)

P $\neq$ NP Proved?

- Razborov's theorem says that there is a monotone language in NP that has no polynomial monotone circuits.
- If we can prove that all monotone languages in P have polynomial monotone circuits, then P $\neq$ NP.
- But Razborov proved in 1985 that some monotone languages in P have no polynomial monotone circuits!

Computation That Counts

And though the holes were rather small,
they had to count them all.
— The Beatles, *A Day in the Life* (1967)

Counting Problems

- Counting problems are concerned with the number of solutions.
  - #SAT: the number of satisfying truth assignments to a boolean formula.
  - #HAMILTONIAN PATH: the number of Hamiltonian paths in a graph.
- They cannot be easier than their decision versions.
  - The decision problem has a solution if and only if the solution count is at least 1.
- But they can be harder than their decision versions.

Decision and Counting Problems

- FP is the set of polynomial-time computable functions $f : \{ 0, 1 \}^* \to \mathbb{Z}$.
  - GCD, LCM, matrix-matrix multiplication, etc.
- If $\#\textsc{sat} \in \mathrm{FP}$, then $\mathrm{P} = \mathrm{NP}$.
  - Given boolean formula ϕ, calculate its number of satisfying truth assignments, k, in polynomial time.
  - Declare "$\phi \in \textsc{sat}$" if and only if $k \geq 1$.
- The validity of the reverse direction is open.

A Counting Problem Harder than Its Decision Version

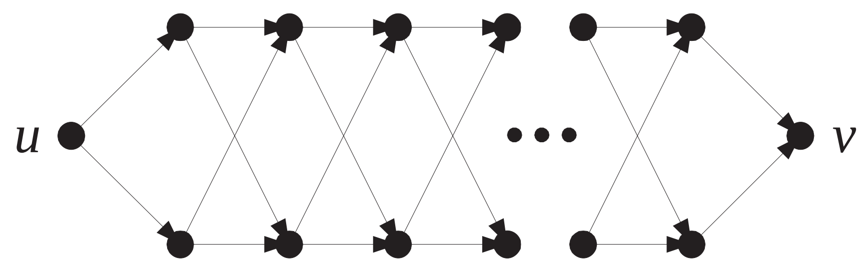
- CYCLE asks if a directed graph contains a cycle.[a]
- #CYCLE counts the number of cycles in a directed graph.
- CYCLE is in P by a simple greedy algorithm.
- But #CYCLE is hard unless P = NP.

[a] A cycle has no repeated nodes.

Hardness of #CYCLE

Theorem 95 (Arora, 2006) *If #CYCLE $\in FP$, then $P = NP$.*

- It suffices to reduce the NP-complete HAMILTONIAN CYCLE to #CYCLE.
- Consider a *directed* graph G with n nodes.
- Define $N \equiv \lfloor n \log_2(n+1) \rfloor$.
- Replace each edge $(u, v) \in G$ with this subgraph:



The Proof (continued)

- This subgraph has $N + 1$ levels.
- There are now 2^N paths from u to v.
- Call the resulting digraph G'.
- Recall that a Hamiltonian cycle on G contains n edges.
- To each Hamiltonian cycle on G, there correspond $(2^N)^n = 2^{nN}$ cycles (not necessarily Hamiltonian) on G'.
- So if G contains a Hamiltonian cycle, then G' contains at least 2^{nN} cycles.

The Proof (continued)

- Now suppose G contains no Hamiltonian cycles.
- Then every cycle on G contains at most $n-1$ nodes.
- There are hence at most n^{n-1} cycles on G.
- Each k-node cycle on G induces $(2^N)^k$ cycles on G'.
- So G' contains at most $n^{n-1}(2^N)^{n-1}$ cycles.
- As $n \geq 1$,

$$
\begin{aligned}
n^{n-1}(2^N)^{n-1} &= 2^{nN} \frac{n^{n-1}}{2^N} \leq 2^{nN} \frac{n^{n-1}}{2^{n \log_2(n+1)-1}} \\
&= 2^{nN} \frac{2n^{n-1}}{(n+1)^n} \leq 2^{nN} \frac{2}{n+1} \left(\frac{n}{n+1}\right)^{n-1} < 2^{nN}.
\end{aligned}
$$

The Proof (concluded)

- In summary, $G \in$ HAMILTONIAN CYCLE if and only if G' contains at least 2^{nN} cycles.
- G' contains at most $n^n \, 2^{nN}$ cycles.
  - Every cycle on G' is associated with a unique cycle on G.
  - Every k-cycle on G induces $(2^N)^k \leq 2^{nN}$ cycles on G'.
  - There are at most n^n cycles in G.
- This number has a polynomial length $O(n^2 \log n)$.
- Hence HAMILTONIAN CYCLE $\in$ P.

Counting Class #P

A function f is in #P (or $f \in \#\mathrm{P}$) if

- There exists a polynomial-time NTM M.
- $M(x)$ has $f(x)$ accepting paths for all inputs x.

Some #P Problems

- $f(\phi)$ = number of satisfying truth assignments to ϕ.
  - The desired NTM guesses a truth assignment T and accepts ϕ if and only if $T \models \phi$.
  - Hence $f \in \#\mathrm{P}$.
  - f is also called #SAT.
- #HAMILTONIAN PATH.
- #3-COLORING.

#P Completeness

- Function f is #P-complete if
  - $f \in$ #P.
  - #P $\subseteq \text{FP}^f$.
    - ∗ Every function in #P can be computed in polynomial time with access to a black box[a] for f.
      - · It said to be polynomial-time Turing-reducible to f.
  - Oracle f can be accessed only a polynomial number of times.

[a]Think of it as a subroutine. It is also called an **oracle**.

#SAT Is #P-Complete[a]

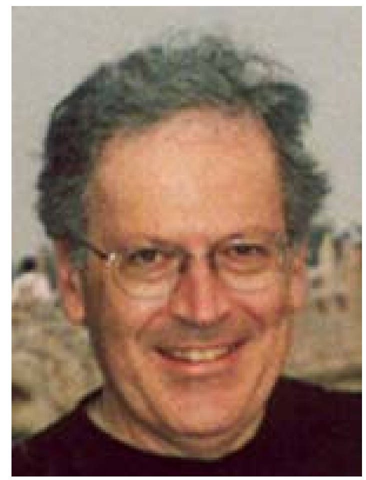
- First, it is in #P (p. 866).
- Let $f \in$ #P compute the number of accepting paths of M.
- Cook's theorem uses a **parsimonious** reduction from M on input x to an instance ϕ of SAT.
  - That is, $M(x)$'s number of accepting paths equals ϕ's number of satisfying truth assignments.
- Call the oracle #SAT with ϕ to obtain the desired answer regarding $f(x)$.

---

[a]Valiant (1979); in fact, #2SAT is also #P-complete.

Leslie G. Valiant[a] (1949–)

Avi Wigderson (2009), "Les Valiant singlehandedly created, or completely transformed, several fundamental research areas of computer science. [...] We all became addicted to this remarkable throughput, and expect more."



[a]Turing Award (2010).