NAESAT

- The NAESAT (for "not-all-equal" SAT) is like 3SAT.
- But there must be a satisfying truth assignment under which no clauses have the three literals equal in truth value.
- Equivalently, there is a truth assignment such that each clause has one literal assigned true and one literal assigned false.

NAESAT Is NP-Complete[a]

- Recall the reduction of CIRCUIT SAT to SAT on p. 231.
- It produced a CNF ϕ in which each clause has at most 3 literals.
- Add the same variable z to all clauses with fewer than 3 literals to make it a 3SAT formula.
- Goal: The new formula $\phi(z)$ is NAE-satisfiable if and only if the original circuit is satisfiable.

[a]Karp (1972).

The Proof (continued)

- Suppose T NAE-satisfies $\phi(z)$.
  - $\bar{T}$ also NAE-satisfies $\phi(z)$.
  - Under T or $\bar{T}$, variable z takes the value false.
  - This truth assignment $\mathcal{T}$ must still satisfy all clauses of ϕ.
    * Note that $\mathcal{T} \models \phi$ with z being false.
  - So it satisfies the original circuit.

The Proof (concluded)

- Suppose there is a truth assignment that satisfies the circuit.
  - Then there is a truth assignment T that satisfies every clause of ϕ.
  - Extend T by adding $T(z) = \texttt{false}$ to obtain T'.
  - T' satisfies $\phi(z)$.
  - So in no clauses are all three literals false under T'.
  - Under T', in no clauses are all three literals true.
    - Need to review the detailed construction on p. 232 and p. 233.

Richard Karp[a] (1935–)

[a]Turing Award (1985).

Undirected Graphs

- An **undirected graph** $G = (V, E)$ has a finite set of nodes, V, and a set of *undirected* edges, E.
- It is like a directed graph except that the edges have no directions and there are no self-loops.
- Use $[\, i, j\,]$ to denote the fact that there is an edge between node i and node j.

Independent Sets

- Let $G = (V, E)$ be an undirected graph.
- $I \subseteq V$.
- I is **independent** if whenever $i, j \in I$, there is no edge between i and j.
- The INDEPENDENT SET problem: Given an undirected graph and a goal K, is there an independent set of size K?
  - Many applications.

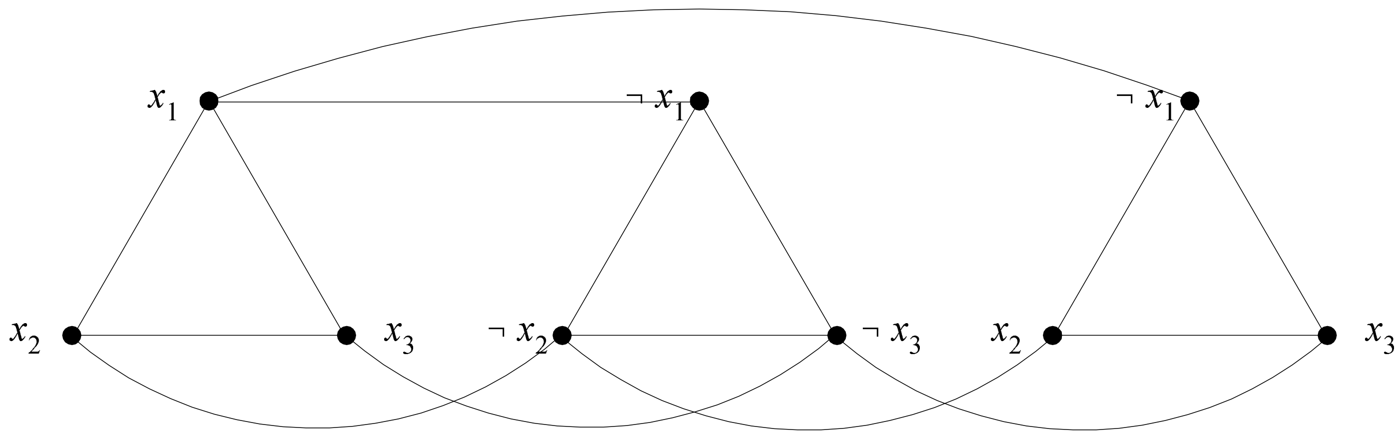
INDEPENDENT SET Is NP-Complete

- This problem is in NP: Guess a set of nodes and verify that it is independent and meets the count.
- If a graph contains a triangle, any independent set can contain at most one node of the triangle.
- The results of the reduction will be graphs whose nodes can be partitioned into m disjoint triangles.
- We will reduce 3SAT to INDEPENDENT SET.

The Proof (continued)

- Let ϕ be an instance of 3SAT with m clauses.
- We will construct graph G (with constraints as said) with $K = m$ such that ϕ is satisfiable if and only if G has an independent set of size K.
- There is a triangle for each clause with the literals as the nodes.
- Add additional edges between x and $\neg x$ for every variable x.

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$



Same literals that appear in different clauses are on distinct nodes.

The Proof (continued)

- Suppose G has an independent set I of size $K = m$.
  - An independent set can contain at most m nodes, one from each triangle.
  - An independent set of size m exists if and only if it contains exactly one node from each triangle.
  - Truth assignment T assigns true to those literals in I.
  - T is consistent because contradictory literals are connected by an edge; hence both cannot be in I.
  - T satisfies ϕ because it has a node from every triangle, thus satisfying every clause.[a]

[a] The variables without a truth value can be assigned arbitrarily. Contributed by Mr. Chun-Yuan Chen (R99922119) on November 2, 2010.

The Proof (concluded)

- Suppose a satisfying truth assignment T exists for ϕ.
  - Collect one node from each triangle whose literal is true under T.
  - The choice is arbitrary if there is more than one true literal.
  - This set of m nodes must be independent by construction.
    - Both literals x and $\neg x$ cannot be assigned true.

Other INDEPENDENT SET-Related NP-Complete Problems

Corollary 36 INDEPENDENT SET *is NP-complete for 4-degree graphs.*

Theorem 37 INDEPENDENT SET *is NP-complete for planar graphs.*

Theorem 38 (Garey and Johnson (1977)) INDEPENDENT SET *is NP-complete for 3-degree planar graphs.*

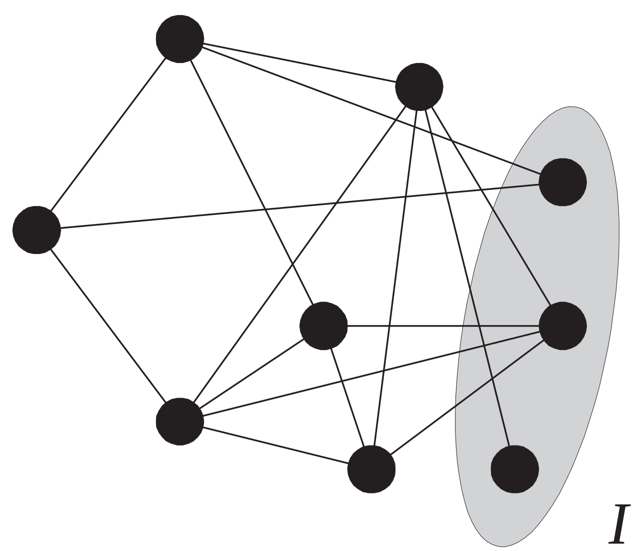
NODE COVER

- We are given an undirected graph G and a goal K.
- NODE COVER: Is there a set C with K or fewer nodes such that each edge of G has at least one of its endpoints in C?

NODE COVER Is NP-Complete

Corollary 39 NODE COVER *is NP-complete.*

- I is an independent set of $G = (V, E)$ if and only if $V - I$ is a node cover of G.



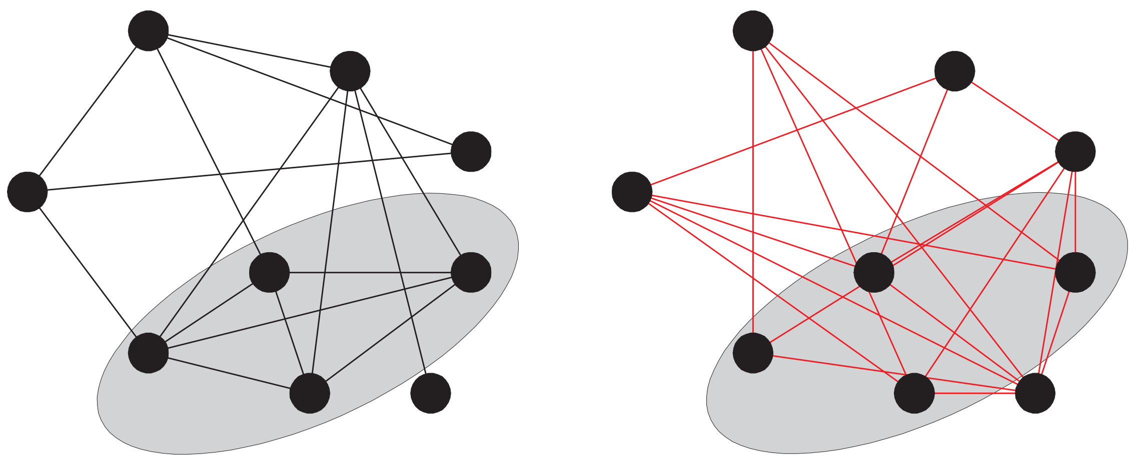
CLIQUE

- We are given an undirected graph G and a goal K.
- CLIQUE asks if there is a set C with K nodes such that whenever $i, j \in C$, there is an edge between i and j.

CLIQUE Is NP-Complete

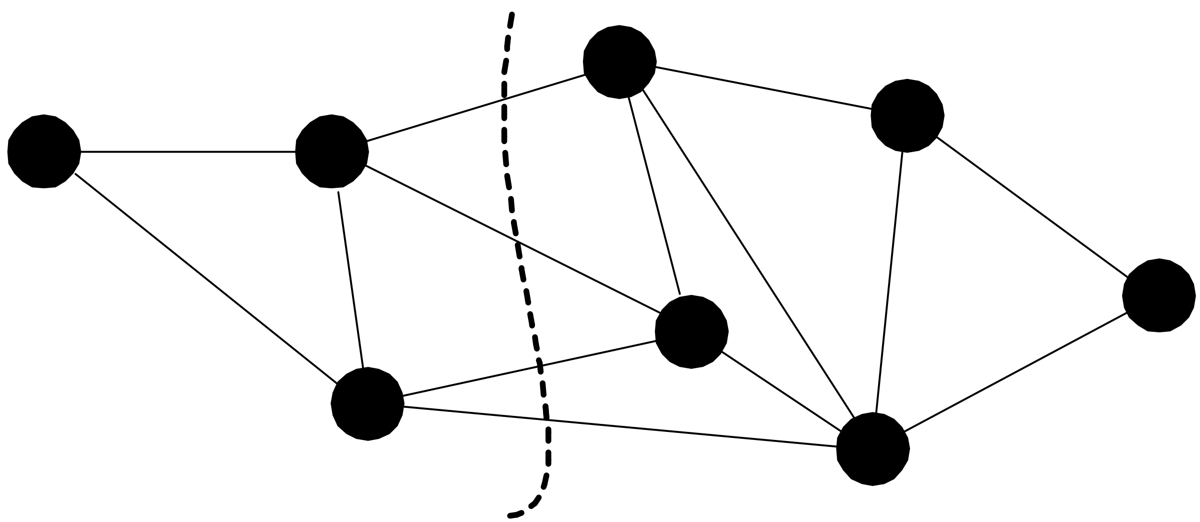
Corollary 40 CLIQUE *is NP-complete.*

- Let $\bar{G}$ be the **complement** of G, where $[x, y] \in \bar{G}$ if and only if $[x, y] \notin G$.
- I is a clique in $G \Leftrightarrow I$ is an independent set in $\bar{G}$.



MIN CUT and MAX CUT

- A **cut** in an undirected graph $G = (V, E)$ is a partition of the nodes into two nonempty sets S and $V - S$.
- The size of a cut $(S, V - S)$ is the number of edges between S and $V - S$.
- MIN CUT $\in$ P by the maxflow algorithm.
- MAX CUT asks if there is a cut of size at least K.
  - K is part of the input.



MIN CUT and MAX CUT (concluded)

- MAX CUT has applications in VLSI layout.
  - The minimum area of a VLSI layout of a graph is not less than the square of its maximum cut size.[a]

[a] Raspaud, Sýkora, and Vrťo (1995); Mak and Wong (2000).

MAX CUT Is NP-Complete[a]

- We will reduce NAESAT to MAX CUT.
- Given an instance ϕ of 3SAT with m clauses, we shall construct a graph $G = (V, E)$ and a goal K such that:
  - There is a cut of size at least K if and only if ϕ is NAE-satisfiable.
- Our graph will have multiple edges between two nodes.
  - Each such edge contributes one to the cut if its nodes are separated.

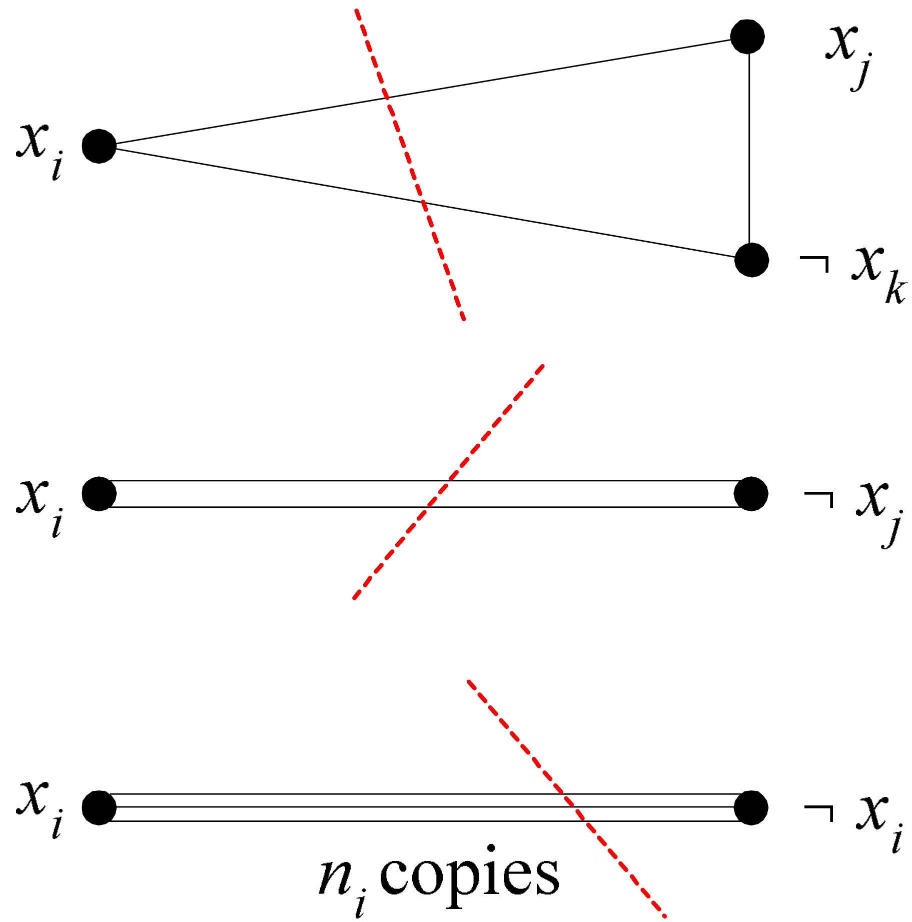
[a]Garey, Johnson, and Stockmeyer (1976).

The Proof

- Suppose ϕ's m clauses are $C_1, C_2, \ldots, C_m$.
- The boolean variables are $x_1, x_2, \ldots, x_n$.
- G has $2n$ nodes: $x_1, x_2, \ldots, x_n, \neg x_1, \neg x_2, \ldots, \neg x_n$.
- Each clause with 3 distinct literals makes a triangle in G.
- For each clause with two identical literals, there are two parallel edges between the two distinct literals.
- No need to consider clauses with one literal (why?).
- For each variable x_i, add n_i copies of edge $[x_i, \neg x_i]$, where n_i is the number of occurrences of x_i and $\neg x_i$ in ϕ.[a]

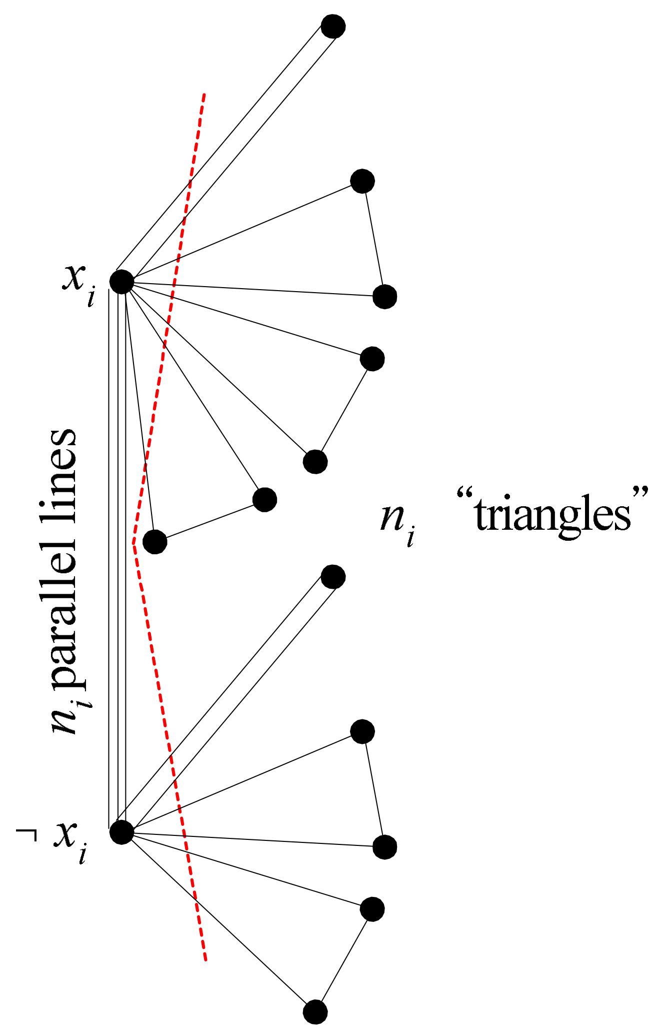
---

[a]Regardless of whether both x_i and $\neg x_i$ occur in ϕ.


x_i
x_j
$\neg x_k$
x_i
$\neg x_j$
x_i
$\neg x_i$
n_i copies

The Proof (continued)

- Set $K = 5m$.
- Suppose there is a cut $(S, V - S)$ of size $5m$ or more.
- A clause (a triangle or two parallel edges) contributes at most 2 to a cut no matter how you split it.
- Suppose both x_i and $\neg x_i$ are on the same side of the cut.
- Then they *together* contribute at most $2n_i$ edges to the cut.
  - They appear in at most n_i different clauses.
  - A clause contributes at most 2 to a cut.



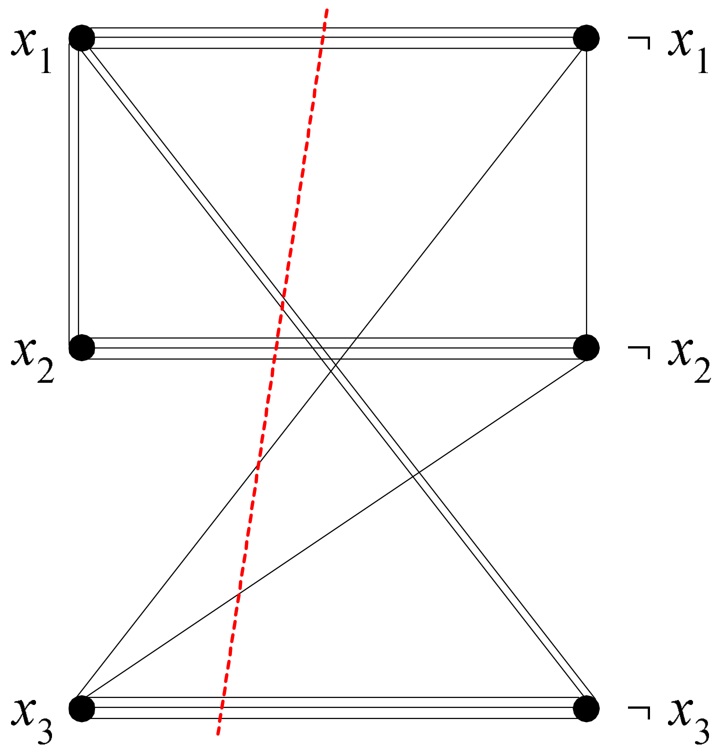
x_i
n_i parallel lines
n_i "triangles"
$\neg\ x_i$

The Proof (continued)

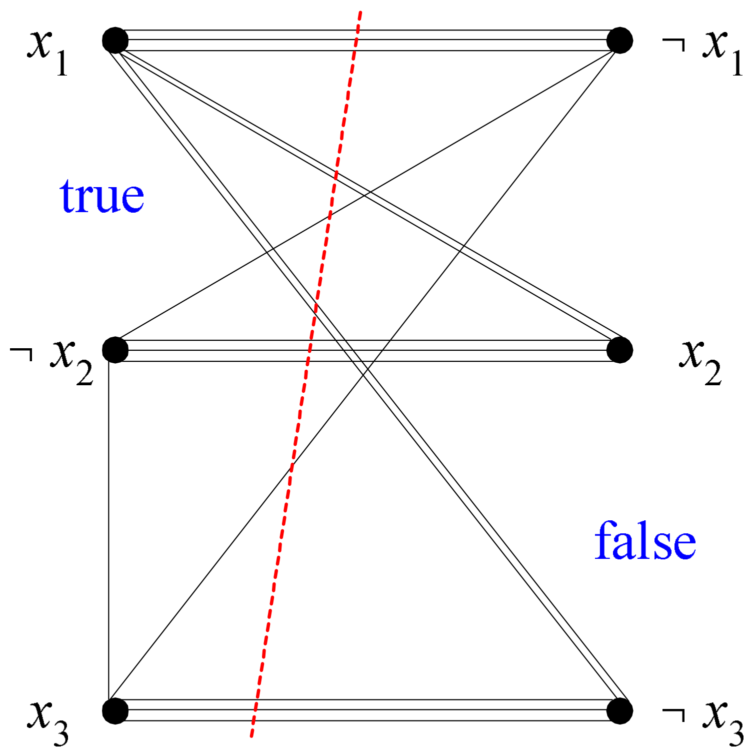
- Either x_i or $\neg x_i$ contributes at most n_i to the cut by the pigeonhole principle.
- Changing the side of that literal does not decrease the size of the cut.
- Hence we assume variables are separated from their negations.
- The total number of edges in the cut that join opposite literals is $\sum_i n_i = 3m$.
  - $\sum_i n_i = 3m$ is the total number of literals.

The Proof (concluded)

- The *remaining* $2m$ edges in the cut must come from the m triangles or parallel edges that correspond to the clauses.
- Each can contribute at most 2 to the cut.
- So all are split.
- A split clause means at least one of its literals is true and at least one false.
- The other direction is left as an exercise.



- $(x_1 \lor x_2 \lor x_2) \land (x_1 \lor \neg x_3 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)$.
- The cut size is $13 < 5 \times 3 = 15$.



- $(x_1 \vee x_2 \vee x_2) \wedge (x_1 \vee \neg x_3 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$.
- The cut size is now 15.

Remarks

- We had proved that MAX CUT is NP-complete for multigraphs.
- How about proving the same thing for simple graphs?[a]
- For 4SAT, how do you modify the proof?[b]
- All NP-complete problems are mutually reducible by definition as an NP-complete problem is in NP.[c]
  - So they are equally hard in this sense.[d]

---

[a] Contributed by Mr. Tai-Dai Chou (J93922005) on June 2, 2005.
[b] Contributed by Mr. Chien-Lin Chen (J94922015) on June 8, 2006.
[c] Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.
[d] Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.