Outline

- NP-complete problems: NAE-3-SAT, max cut
- The complexity class coNP
- The complexity class coNP ∩ NP

Outline

- NP-complete problems: NAE-3-SAT, max cut
- The complexity class coNP
- The complexity class coNP ∩ NP

Outline

- NP-complete problems: NAE-3-SAT, max cut
- The complexity class coNP
- The complexity class coNP ∩ NP

Not-All-Equal 3SAT

\[(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_4 \lor x_5) \land \ldots \land (\ldots)\]

**Theorem:** the following language is NP-complete:

NAE3SAT = \{φ : φ is a 3-CNF formula for which there exists a truth assignment in which every clause has at least 1 true literal and at least 1 false literal\}

- Proof:
  - Part 1: NAE3SAT ∈ NP. Proof?
  - Part 2: NAE3SAT is NP-hard. Reduce from?

NAE3SAT is NP-complete

- Recall reduction to 3SAT
  - variables \(x_1, x_2, \ldots, x_n\), gates \(g_1, g_2, \ldots, g_m\)
  - produce clauses:
    - \((\neg z_1 \lor \neg z_2 \lor v)\)
    - \((\neg g \lor z_1 \lor z_2)\)
    - \((\neg g \lor z_1)\)
    - \((\neg z_1 \lor \neg z_2 \lor v)\)
    - \((g \lor z)\)
    - \((\neg g \lor z_1 \lor z_2)\)
    - \((\neg g \lor z_1)\)
    - \((\neg z_1 \lor \neg z_2 \lor v)\)

    output gate \(g_m\):

    - \((g_m \lor w)\)

    not all true in a satisfying assignment
NAE3SAT is NP-complete

- Does the reduction run in polynomial time?
  - \((\neg z_1 \lor g \lor w)\)
  - \((\neg z_2 \lor g \lor w)\)
  - \((\neg g \lor z_1 \lor z_2)\)
  - \((\neg g \lor z_1 \lor w)\)
  - \((\neg g \lor z_2 \lor w)\)
  - \((\neg z_1 \lor \neg z_2 \lor g)\)
  - \((g \lor z \lor w)\)
  - \((\neg z \lor \neg g \lor w)\)
  - \((g \lor z \lor w)\)

- YES maps to YES
  - already know how to get a satisfying assignment to the BLUE variables
  - set \(w = \text{FALSE}\)

\[(\neg z_1 \lor g \lor w)\]
\[(\neg z_2 \lor g \lor w)\]
\[(\neg g \lor z_1 \lor z_2)\]
\[(\neg g \lor z_1 \lor w)\]
\[(\neg g \lor z_2 \lor w)\]
\[(\neg z_1 \lor \neg z_2 \lor g)\]
\[(g \lor z \lor w)\]
\[(\neg z \lor \neg g \lor w)\]
\[(g \lor z \lor w)\]

MAX CUT

- Given graph \(G = (V, E)\)
  - a cut is a subset \(S \subseteq V\)
  - an edge \((x, y)\) crosses the cut if \(x \in S\) and \(y \in V - S\) or \(x \in V - S\) and \(y \in S\)
  - search problem:
    find cut maximizing number of edges crossing the cut

MAX CUT is NP-complete

- We are reducing from the language:
  NAE3SAT = \(\{\phi : \phi\text{ is a 3-CNF formula for which there exists a truth assignment in which every clause has at least 1 true literal and at least 1 false literal}\}\)

  to the language:
  MAX CUT = \(\{(G = (V, E), k) : \text{there is a cut } S \subseteq V \text{ with at least } k \text{ edges crossing it}\}\)
MAX CUT is NP-complete

- The reduction:
  - given instance of NAE3SAT \((n, m)\): 
    \((x_1 \lor x_2 \lor \neg x_3) \land \neg (x_1 \lor x_4 \lor x_5) \land \ldots \land \neg (x_2 \lor x_3 \lor x_4)\)
  - produce graph \(G = (V, E)\) with node for each literal

- triangle for each 3-clause
- parallel edges for each 2-clause

- YES maps to YES
  - take cut to be TRUE literals in a NAE truth assignment
  - contribution from clause gadgets: \(2m\)
  - contribution from \((x_i, \neg x_i)\) parallel edges: \(3m\)

- set \(k = 5m\)

- NO maps to NO
  - Claim: if cut has \(x_i, \neg x_i\) on same side, then can move one to opposite side without decreasing # edges crossing cut
  - Proof:

  \[ a + n_i \leq 2n_i \]

  \[ a + b \leq 2n_i \]

  \[ a + n_i \geq a + b \text{ or } b + n_i \geq a + b \]

coNP

- Is NP closed under complement?
  - Can we transform this machine: 
  - into a machine with this behavior?
coNP

- Language L is in coNP iff its complement (co-L) is in NP

- It is believed that NP ≠ coNP
- Note: P = NP implies NP = coNP
  - Proving NP ≠ coNP would prove P ≠ NP
  - Another major open problem...