CSE 417: Algorithms and Computational Complexity

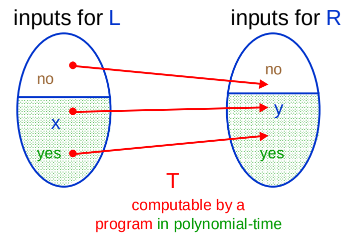
Winter 2001

Lecture 23

Instructor: Paul Beame

1

Polynomial-time reduction L $\le^P$R



L answers yes to x $\Leftrightarrow$ R answers yes to T(x)

2

e.g., Independent-Set $\le^P$Clique

- Define reduction T, that maps <G,k> to <$\bar{G}$,k>, where $\bar{G}$ is the complement graph of G, i.e. $\bar{G}$ has an edge exactly when G doesn't.
  - Clearly polynomial time.
- Correctness
  - <G,k> is a YES for Independent-Set
  - iff there is a subset U of the vertex set of G with $|U| \ge k$ such that no two vertices in U are joined by an edge in G
  - iff there is a subset U of the vertex set of $\bar{G}$ with $|U| \ge k$ such that every pair of vertices in U is joined by an edge in $\bar{G}$
  - iff <$\bar{G}$,k> is a YES for Clique

3

NP-hardness & NP-completeness

- Definition: A problem R is NP-hard iff every problem L$\in$NP satisfies L $\le^P$R
- Definition: A problem R is NP-complete iff R is NP-hard and R $\in$NP
- Not obvious that such problems even exist!

4

Properties of polynomial-time reductions

- Theorem: If L $\le^P$R and R is in P then L is also in P
- Theorem: If L $\le^P$R and R $\le^P$S then L $\le^P$S
- Proof idea:
  - Compose the reduction T from L to R with the reduction T' from R to S to get a new reduction T''(x)=T'(T(x)) from L to S.

5

Cook's Theorem& implications

- Theorem (Cook 1971): Satisfiability is NP-complete
- Corollary: R is NP-hard iff Satisfiability $\le^P$R
  - (or Q $\le^P$R for any NP-complete problem Q)
- Proof:
  - If R is NP-hard then every problem in NP polynomial-time reduces to R, in particular Satisfiability does since it is in NP
  - For any problem L in NP, L $\le^P$Satisfiability and so if Satisfiability $\le^P$R we have L $\le^P$ R.
    - therefore R is NP-hard if Satisfiability $\le^P$R

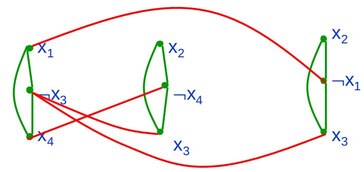
6

Another NP-complete problem: Satisfiability $\le^P$ Independent-Set

- Reduction:
  - mapping CNF formula F to a pair <G,k>
  - Let m be the number of clauses of F
  - Create a vertex in G for each literal in F
  - Join two vertices u, v in G by an edge iff
    - u and v correspond to literals in the same clause of F, (green edges) or
    - u and v correspond to literals x and ¬x (or vice versa) for some variable x. (red edges).
  - Set k=m
  - Clearly polynomial-time

Satisfiability $\le^P$ Independent-Set

F: $(x_1 \lor \neg x_3 \lor x_4) \land (x_2 \lor \neg x_4 \lor x_3) \land (x_2 \lor \neg x_1 \lor x_3)$



Satisfiability $\le^P$ Independent-Set

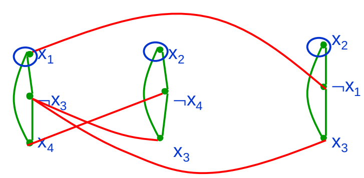
- Correctness:
  - If F is satisfiable then there is some assignment that satisfies at least one literal in each clause.
  - Consider the set U in G corresponding to the first satisfied literal in each clause.
    - |U|=m
    - Since U has only one vertex per clause, no two vertices in U are joined by green edges
    - Since a truth assignment never satisfies both x and ¬x, U doesn't contain vertices labeled both x and ¬x and so no vertices in U are joined by red edges
    - Therefore G has an independent set, U, of size at least m
  - Therefore <G,m> is a YES for independent set.

Satisfiability $\le^P$ Independent-Set

1 0 1 1 0 1 1 0 1

F: $(x_1 \lor \neg x_3 \lor x_4) \land (x_2 \lor \neg x_4 \lor x_3) \land (x_2 \lor \neg x_1 \lor x_3)$

U



Given assignment $x_1=x_2=x_3=x_4=1$,
U is as circled

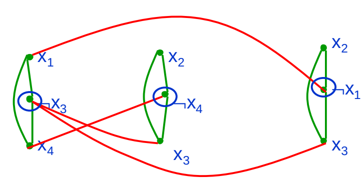
Satisfiability $\le^P$ Independent-Set

- Correctness continued:
  - If <G,m> is a YES for Independent-Set then there is a set U of m vertices in G containing no edge.
    - Therefore U has precisely one vertex per clause because of the green edges in G.
    - Because of the red edges in G, U does not contain vertices labeled both x and ¬x
    - Build a truth assignment A that makes all literals labeling vertices in U true and for any variable not labeling a vertex in U, assigns its truth value arbitrarily.
    - By construction, A satisfies F
  - Therefore F is a YES for Satisfiability.

Satisfiability $\le^P$ Independent-Set

0 1 0 ? 1 0 ? 1 0

F: $(x_1 \lor \neg x_3 \lor x_4) \land (x_2 \lor \neg x_4 \lor x_3) \land (x_2 \lor \neg x_1 \lor x_3)$



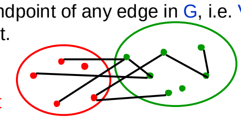
Given U, satisfying assignment
is $x_1=x_3=x_4=0$, $x_2=0$ or 1

Independent-Set is NP-complete

- We just showed that Independent-Set is NP-hard and we already knew Independent-Set is in NP.
- Corollary: Clique is NP-complete
  - We showed already that Independent-Set $\le^P$ Clique and Clique is in NP.

Vertex Cover

- Given an undirected graph G=(V,E) and an integer k is there a subset W of V of size at most k such that every edge of G has at least one endpoint in W? (i.e. W covers all vertices of G).
- Observation:
  - W is a vertex cover of G iff every vertex in V-W has at most one endpoint of any edge in G, i.e. V-W is independent.



Independent-Set $\le^P$ Vertex Cover

- Reduction:
  - Map <G,k> to <G,|V|-k>.
  - Correctness follows from Observation
  - Polynomial-time
- Vertex Cover is in NP
  - Hint is the cover set W.
  - Polynomial-time to check.
- Vertex Cover is NP-complete

Steps to Proving Problem R is NP-complete

- Show R is NP-hard:
  - State:'Reduction is from NP-hard Problem L'
  - Show what the map is
  - Argue that the map is polynomial time
  - Argue correctness: two directions Yes for L implies Yes for R and vice versa.
- Show R is in NP
  - State what hint is and why it works
  - Argue that it is polynomial-time to check.