Lecture 32: Friday April 4, 2003

today

- NP completeness

announcements

- final exam

recall: basic concepts

- (abstract) problem
- decision problem
- the classes P, NP, co-NP
- the satifiability problem (SAT)
- completeness: the class NP-c
- Cook's theorem: SAT is in NP-c
- Karp's consequences: these problems are in NP-c
  - graph 3-colouring
  - graph Hamiltonian
  - ... (10 others)
- today: know thousands of problems $\in$ NP-c
- how to show a problem $\in$ NP-c

recall: polynomial time reduction

- a decision problem Π_1 is *polynomially reducible* to a decision problem Π_2 if there is a polynomial time transformation function t which maps instances of Π_1 to instances of Π_2 s.t. s.t. for all instances x of Π_1, the answer to x is the same as the answer to $t(x)$
- e.g. k-independent set is reducible to k-clique
  proof: X independent set in G iff X clique in $\overline{G}$
- e.g. 3-sat is reducible to sat — trivial exercise
- e.g. sat is reducible to 3-sat — non-trivial exercise

(formula) satisfiability (SAT) [CLRS p996]

- boolean formula (recursive definition)
  - boolean variable x_j
  - $\neg(f)$ where f is a boolean formula
  - $f \wedge g$ where f, g are boolean formulas
  - $f \vee g$ where f, g are boolean formulas
  - $f \rightarrow g$ where f, g are boolean formulas
  - $f \leftrightarrow g$ where f, g are boolean formulas
- satisfiable if, for some boolean assignment, formala evaluates TRUE
- $(\ (x_1 \rightarrow x_2) \vee \neg(\ (\neg x_1 \leftrightarrow x_3) \vee x_4)\ ) \wedge \neg x_2$
- above formula satisfiable: $x_1 = F\ x_2 = F\ x_3 = T\ x_4 = T$
- satisfiability problem (SAT)
  - instance: boolean formula
  - query: is the formula satisfiable?

conjunctive normal form satisfiability (CNF-SAT)

- literal: x_j or $\neg x_j$
- boolean CNF formula: $c_1 \wedge c_2 \wedge \ldots \wedge c_m$

  where each clause c_j is the OR of one or more literals
- $(\neg x_3 \vee x_7) \;\wedge\; (x_{11}) \;\wedge\; (\neg x_2 \vee x_3 \vee \neg x_5 \vee x_6) \;\wedge\; (x_2 \vee x_3 \vee x_8)$
- CNF-SAT problem:
  - instance: boolean CNF formula
  - query: is the formula satisfiable?

3-CNF-SAT

- 3-CNF: CNF, and each clause has exactly 3 distinct literals
- $(\neg x_3 \vee x_7 \vee x_8) \wedge (x_4 \vee x_6 \vee x_9) \wedge (\neg x_2 \vee x_3 \vee x_6) \wedge (x_2 \vee x_3 \vee x_8)$
- 3-CNF-SAT problem:
  - instance: boolean 3-CNF formula
  - query: is the formula satisfiable?

the class of problems NP-c

- NP-complete problem
  - is in NP
  - *every* problem in NP is polynomially reducible to it
- NP-c: all NP-complete problems

Cook's theorem [about 1971]

- SAT $\in$ NP-c

the classes P, NP, co-NP, NP-c

- if any NP-complete problem is in P, then *every* problem in NP is in P (not likely)
- least likely: P=NP=co-NP
- picture:

- most likely: P, NP, co-NP, NP-c all different
- picture:

big open CS problem **P = NP?**

Karp's consequences of Cook's theorem

- recall: Cook showed SAT in NP-c
- soon after: Karp showed 21 other problems in NP-c
- how did Karp do this so quickly?
- using Cook's idea of polynomial reduction
- from Karp's paper:
  - SAT reduces to 3-CNF-SAT
  - CNF-SAT reduces to 3-colouring
  - 3-colouring reduces to k-colouring (fixed $k \geq 3$)
  - SAT reduces to k-clique
  - SAT reduces to k-independent set
  - ...
- now: know thousands of problems in NP-c

how to show that a problem B is in NP-c

- use polynomial reduction (transformation)
  - show that B is in NP — usually easy
  - find an 'old' problem A (known to be in NP-c) — e.g. SAT
  - poly'ly reduce A to B
- consequence
  - any problem in NP can be reduced into A
  - A can be reduced into B . . . so
  - any problem in NP can be reduced into B . . . so
  - B is in NP-c
  - picture:

examples of polynomial reductions

- we will show
  - CNF-SAT reduces to 3-CNF-SAT
  - 3-CNF-SAT reduces to 3-colouring
  - 3-colouring reduces to 4-colouring

example reduction: CNF-SAT to 3-CNF-SAT

- transformation input: CNF-SAT formula f
- transformation output: 3-CNF-SAT formula f', such that
  f satisfiable iff f' satisfiable
- to obtain f' from f, for each clause

1 literal: $(z) \rightarrow$

$$(z \vee p \vee q) \wedge (z \vee p \vee \neg q) \wedge (z \vee \neg p \vee q) \wedge (z \vee \neg p \vee \neg q)$$

2 literals: $(y \vee z) \rightarrow$

$$(y \vee z \vee p) \wedge (y \vee z \vee \neg p)$$

3 literals: $(x \vee y \vee z) \rightarrow$

$$(x \vee y \vee z)$$

4 literals: $(v_1 \vee v_2 \vee v_3 \vee v_4) \rightarrow$

$$(v_1 \vee v_2 \vee p_1) \wedge (\neg p_1 \vee v_3 \vee v_4)$$

5 literals: $(v_1 \vee v_2 \vee v_3 \vee v_4 \vee v_5) \rightarrow$

$$(v_1 \vee v_2 \vee p_1) \wedge (\neg p_1 \vee v_3 \vee p_2) \wedge (\neg p_2 \vee v_4 \vee v_5)$$

...

consequence

- since
  - CNF-SAT is in NP-c (Cook proved this)
  - 3-CNF-SAT is in NP
  - CNF-SAT is polynomial-time-reducible to 3-CNF-SAT
- it follows that 3-CNF-SAT is in NP-c