CSC196: Great Ideas in Computing

Tutorial 6

18th November 2022

Any questions from Assignment 3?

- You still have grace period of 2 days with 0.5% penalty each hour

Polynomial time transformations

- **NP:** Problems that can be solved in a polynomial time by a non-deterministic machine.
  - In other words, given a certificate/solution, it can be verified in polynomial time
  - e.g. Weighted independent set
- **NP complete:** If problem A is NP complete, then any NP problem B can be reduced in polynomial time to problem A (i.e. $B \leq_P A$)
  - If we can solve a NP complete problem in polynomial time then $P = NP$
  - e.g. 3-SAT problem

Polynomial time transformations (cont.)

- **NP hard:** These problems are at least as difficult as NP complete problems.
    - If problem A is NP hard, then there exists a NP-complete problem B which is reducible to A in polynomial time.

Relationship between Independent Set and Vertex cover

- **Independent set problem:** Given a graph $G = (V, E)$ and a number k, does G contain a set of at least k independent vertices (not connected to each other)?
- **Vertex cover:** A vertex cover S of a graph G is a subset of nodes such that every edge has at least one endpoint in S
  - In other words, we try to "cover" each edge by choosing at least one of its vertices

Relationship between Independent Set and Vertex cover (cont.)

- **Vertex cover problem:** Given a graph $G = (V, E)$ and a number k, does G contain a vertex cover of size at most k?
- **Independent Set** $\leq_P$ **Vertex cover**
  - Given an instance of independent set $< G, k >$
  - We ask our Vertex Cover black box if there is a vertex cover $V - S$ of size $|V| - k$
    - If yes, then S is an independent set of size $\geq k$
    - If no, then there is no independent set of size $\geq k$

Relationship between Independent Set and Vertex cover (cont.)

- We also have **Vertex cover** $\leq_P$ **Independent set**
- So, vertex cover and independent set problems are equivalently difficult
- We know that both are NP-complete

Set Cover problem

- Given a set U of elements and a collection $S_1, \ldots, S_n$ subsets of U, is there a collection of **at most** k of these subsets whose union equal U?
- Vertex Cover $\leq_P$ Set Cover
  - Let $< G = (V, E), k >$ be an instance of Vertex Cover problem
  - Create an instance of Set Cover problem
    - $U = E$
    - Create S_u for each $u \in V$ where S_u contains all the edges adjacent to u
- U can be covered by $\leq k$ set iff G has a vertex cover of size $\leq k$

Set Cover problem (cont.)

- Set Cover problem is NP-Complete
  - We know that Vertex Cover problem is NP-Complete
  - Vertex Cover $\leq_P$ Set Cover
  - Any certificate/solution of Set Cover problem can be verified in polynomial time

Propositional logic

- Required for Assignment 4
- DeMorgan's Law
  - $\neg(P \vee Q) \equiv (\neg P \wedge \neg Q)$
  - $\neg(P \wedge Q) \equiv (\neg P \vee \neg Q)$
- Notation
  - P, Q are boolean variables which can take values True or False
  - $\vee$ is **OR** condition
  - $\wedge$ is **AND** condition
  - $\equiv$ is equivalence condition

Propositional logic (cont.)

- $P \equiv Q$
  - $P \Rightarrow Q$
  - $Q \Rightarrow P$
- Material Implication
  - $P \Rightarrow Q \equiv (\neg P \vee Q)$
- Distribution
  - $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$
  - $a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c)$

Propositional logic (cont.)

- Double negation
  - $\neg\neg a \equiv a$

Conjunctive Normal Form (CNF)

- A boolean expression is said to be in CNF
  - It is a conjunction ($\wedge$) of clauses
  - Each clause is a disjunction ($\vee$) of literals
  - Each literal is a variable (a) or negation of a variable ($\neg a$)
- Using propositional logic any boolean expression can be expressed in CNF
- $k - CNF$ means every clause should have a maximum of k literals

SAT

- Is there an allocation of variables such that a boolean expression is **satisfiable**
- **3-SAT**: Every clause has at most 3 variables
- $k - SAT$ can be reduced to $(k-1) - SAT$ in polynomial time for $k >= 3$
  - If we keep going on, we can reduce $k - SAT$ problem in polynomial time to $3 - SAT$ problem

SAT (cont.)

- Cook-Levin theorem: SAT is NP-complete
- So, 3-SAT is also NP-complete due to aforementioned reduction
  - Note the direction of reduction

References

- NP vs NP-complete vs NP-hard
  - https://stackoverflow.com/questions/1857244/what-are-the-differences-between-np-np-complete-and-np-hard
- Vertex Cover, Independent Set and Set Cover
  - https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/npcomplete.pdf
- Weighted Independent set
  - https://cs.stackexchange.com/questions/92914/prove-that-weighted-independent-set-is-np-complete-using-independent-set
- Propositional Logic
  - https://iep.utm.edu/prop-log/

References

- SAT, 3-SAT, CNF, Cook-Levinson theorem
  - https://www.baeldung.com/cs/cook-levin-theorem-3sat