CMPSCI 311: Introduction to Algorithms

More Reductions, NP

Dan Sheldon

University of Massachusetts

Last Compiled: April 24, 2017

Reduction Strategies

- Reduction by equivalence (Vertex Cover and Indpendent Set)
- Reduction to a more general case
- Reduction by "gadgets"

Reduction by Gadgets: Satisfiability

- Can we determine if a Boolean formula has a satisfying assignment?

$$(x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_2 \vee \bar{x}_3)$$

- Boolean variables: $x_1, \dots, x_n$
- Term: a variable or its negation. x_i or $\bar{x}_i$
- Clause: a disjunction ("or") of terms. $C = x_1 \vee \bar{x}_2 \vee x_4$
- Formula: a conjunction ("and") of clauses. $C_1 \wedge C_2 \wedge C_3 \wedge C_4$
- Assignment: assign 0/1 to each variable. $x_1 = 1, x_2 = 1, x_3 = 1$
- Satisfying assignment: makes all clauses evaluate to "true". $x_1 = 0, x_2 = 0, x_3 = 0$

Reduction by Gadgets: Satisfiability

SAT – Given boolean formula $C_1 \wedge C_2 \dots \wedge C_m$ over variables $x_1, \dots, x_n$, does there exist a satisfying assignment?

3-SAT – Same, but each C_i has exactly three terms

Claim: 3-SAT $\leq_P$ IndependentSet.

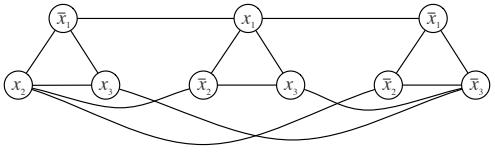
Reduction:

- Given 3-SAT instance $\Phi = \langle C_1, \dots, C_m \rangle$, we will construct an independent set instance $\langle G, m \rangle$ such that G has an independent set of size m iff Φ is satisfiable
- Return Yes if `solveIS`$(\langle G, m \rangle)$ = Yes

Reduction

- **Idea**: construct graph G where independent set will select one term per clause to be true

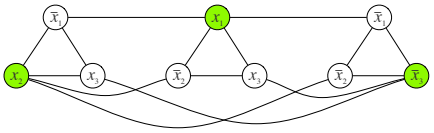
$$(\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$$



- One node per term
- Edges between all terms in same clause (select at most one)
- Edges between a literal and all of its negations (consistent truth assignment)

Correctness

$$(\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$$

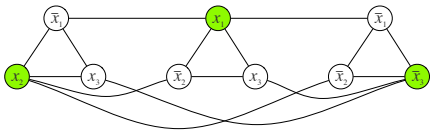


Claim: if G has an independent set of size m, then $\langle C_1, \dots, C_m \rangle$ is satisfiable

- Suppose S is an independent set of size m
- Assign variables so selected literals are true. Edges from terms to negations ensure non-conflicting assignment.
- Set any remaining variables arbitrarily
- At most one term per clause is selected. Since m are selected, every clause is satisfied.

Correctness

$(\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$

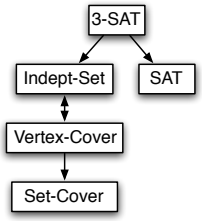


Claim: if $\langle C_1, \ldots, C_m \rangle$ is satisfiable, then G has an independent set of size m

- Consider any satsifying assignment of $\langle C_1, \ldots, C_m \rangle$
- Let S consist of one node per triangle corresponding to true literal in that clause. Then $|S| = m$.
- For (u, v) within clause, at most one endpoint is selected
- For edge $(x_i, \bar{x}_i)$ between clauses, at most one endpoint is selected, because $x_i = 1$ or $\bar{x}_i = 1$, but not both
- Therefore S is an independent set

Reductions So Far

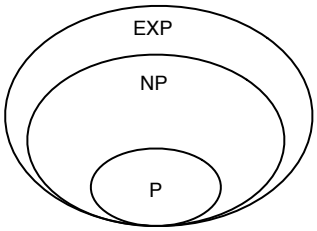
Partial map of problems we can use to solve others in polynomial time, through transitivity of reductions:



- $\boxed{Y} \longrightarrow \boxed{X}$ means $Y \leq_P X$.

Toward a Definition of NP

Remember our mystery problems:



What is special about these?

P and NP

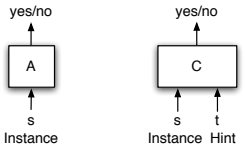
- P: Decision problems for which there is a polynomial time algorithm.
- NP: Decision problems for which there is a polynomial time certifier.

Intuition: A correct solution can be certified in polynomial time.

Solver vs. Certifier

Let X be a decision problem and s be problem instance (e.g., $s = \langle G, k \rangle$ for Independent Set)

Poly-time solver. Algorithm $A(s)$ such that $A(s) = \text{Yes}$ iff correct answer is Yes, and running time polynomial time in $|s|$



Poly-time certifier. Algorithm $C(s, t)$ such that for every instance s, there is some t such that $C(s, t) = \text{Yes}$ t iff correct answer is Yes, and running time is polynomial in $|s|$.

- t is the "certificate" or hint. Size of t must also be polynomial in $|s|$

Certifier Example: Independent Set

Input $s = \langle G, k \rangle$.
Problem: Does G have an independent set of size at least k?
Idea: Certificate t = an independent set of size k

```
CertifyIS( ⟨G, k⟩, t)
  if |t| < k return No
  for each edge e = (u, v) ∈ E do
      if u ∈ t and v ∈ t return No
  end for
  Return Yes
```

Polynomial time?

Example: Independent Set

- INDEPENDENT SET $\in$ P?
  - Unknown. No known polynomial time algorithm.
- INDEPENDENT SET $\in$ NP?
  - Yes. Easy to certify solution in polynomial time.

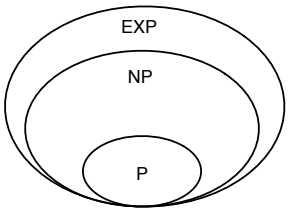
Example: 3-SAT

Input: formula Φ on n variables.
Problem: Is Φ satisfiable?
Idea: Certificate t = the satisfying assignment

`Certify3SAT(` $\langle \Phi \rangle, t$)

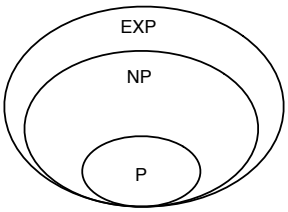
▷ Check if t makes Φ true

Takeaway



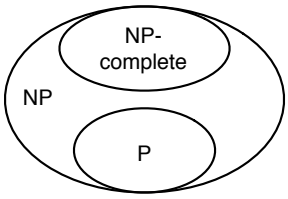
- 3SAT and INDEPENDENT SET are in NP, as are many other problems that are hard to solve, but easy to certify!

P, NP, EXP



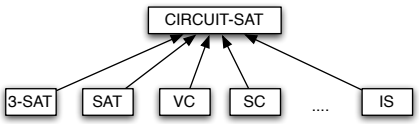
- **Claim**: P $\subseteq$ NP
- **Claim**: NP $\subseteq$ EXP
- Both straightforward to prove, but not critical right now.

NP-Complete



- NP-complete = a problem $Y \in$ NP with the property that $X \leq_P Y$ for every problem $X \in$ NP!

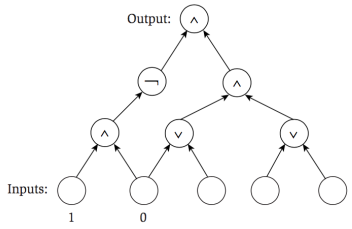
NP-Complete



- **Cook-Levin Theorem**: In 1971, Cook and Levin independently showed that particular problems were NP-Complete.
- We'll look at CIRCUIT-SAT as canonical NP-Complete problem.

Circuit-SAT

Problem: Given a circuit built of And, Or, and Not gates with some hard-coded inputs, is there a way to set remaining inputs so the output is 1?



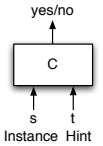
Satisfiable? Yes. Set inputs: 1, 1, 0.

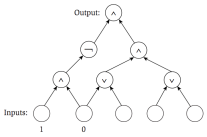
Circuit-SAT

Cook-Levin Theorem Circuit-SAT is NP-Complete.

Proof Idea: encode arbitrary certifier $C(s,t)$ as a circuit

- If $X \in$ NP, then X has a poly-time certifier $C(s,t)$

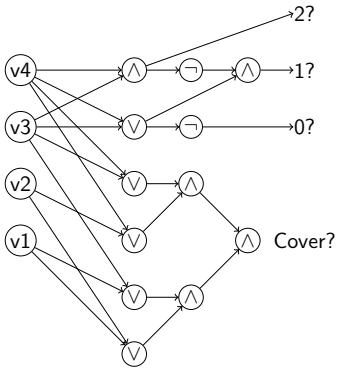




- Construct a circuit where s is hard-coded, and circuit is satsifiable iff $\exists t$ that causes $C(s,t)$ to output Yes
- Algorithm for Circuit-Sat implies an algorithm for X

A Circuit-SAT reduction

- Vertex Cover – Does G have VC of size at most k?



Back to 3-SAT

Claim. If Y is NP-complete and $Y \leq_P X$, then X is NP-complete.
Theorem. 3-SAT is NP-Complete.

- Clearly in $\mathcal{NP}$.
- Prove by reduction from CircuitSAT.

Example.