8.9 co-NP and the Asymmetry of NP

NP and co-NP

DEF 1. (NP) Decision problems for which there is a **poly-time certifier:**

X ∈ NP iff ∃ poly-time certifier C(s, t) s.t. for every string $s \in \Sigma^*$,

s ∈ X (i.e. s is a `yes Instance`) iff ∃ a string t such that C(s, t) = `yes`.

t = certificate or witness

Ex. SAT, HAM-CYCLE, COMPOSITES.

Def. Given a decision problem X, its complement X is the same problem with the `yes` and `no` answers **reverse**.

Ex. X = { 0, 1, 4, 6, 8, 9, 10, 12, 14, 15, … } = { `yes` **Instances** of **COMPOSITES**}

Co-X = { 2, 3, 5, 7, 11, 13, 17, 23, 29, … }

co-NP = { All Complements of decision problems in NP}

Ex. Co-SAT (TAUTOLOGY), NO-HAM-CYCLE, PRIMES ∈ Co-NP

Homework: Define Co-X in terms of certifiers and certificates using DEF 1.

Asymmetry of NP

Asymmetry of NP:

- We only need to have **at least one** short proof (Certificate) of `yes` **instances**.
- While, for `no` **instances**, we require **no** (short) proof must exist.

Ex 1. **SAT vs. Co-SAT (TAUTOLOGY).**

- **SAT:** Can prove a **CNF formula** is **satisfiable** by giving **good assignment (certificate)**.
- **Co-SAT**: How could we prove that a **CNF formula** is **not** satisfiable via a **short certificate**?

Ex 2. **HAM-CYCLE vs. Co-HAM-CYCLE.**

- Can prove a graph is **Hamiltonian** by giving such a **Hamiltonian cycle**.
- How could we prove that a graph is **not Hamiltonian** via a **short certificate**?

Remark. SAT is **NP-complete** and SAT $\equiv_P$ **Co-SAT (Homework)**,

....but....

↑

how do we classify **Co-SAT** ? Is it in NP? can be in **P?**

We don't even know whether it is in NP !

NP = co-NP ?

Fundamental question. Does NP = co-NP?

- Do yes instances have succinct certificates iff no instances do?
- Consensus opinion: no!

Theorem. If NP $\neq$ co-NP, then P $\neq$ NP.

Proof (By Contradiction).

- P is **closed** under complementation (i.e. P=Co-P) (do as Homework).
- If P = NP, then NP is also **closed** under complementation.
- In other words, NP = co-NP.
- Contradiction!

Good Characterizations

Good characterization. [Edmonds 1965]

What is NP ∩ co-NP ?

- If problem **X** is in both NP and co-NP, then:
  - for `yes` **instance**, there is a short certificate t^{YES}
  - for `no` **instance**, there is a short **disqualifier** t^{NO}

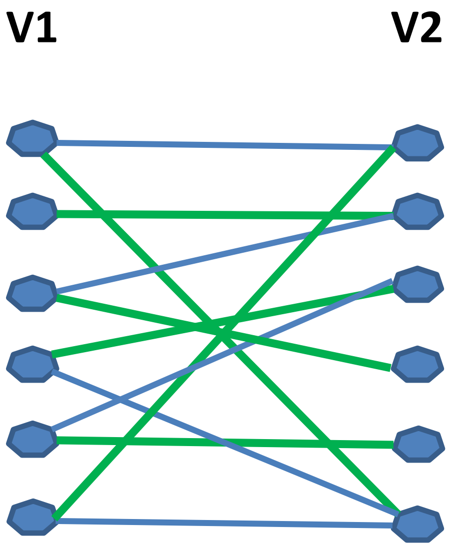
Ex. Given a **bipartite graph G(V,E)**, is there a Perfect Matching.

- If yes, can exhibit a perfect matching.
- If no, can exhibit a set of nodes S such that $|N(S)| < |S|$.

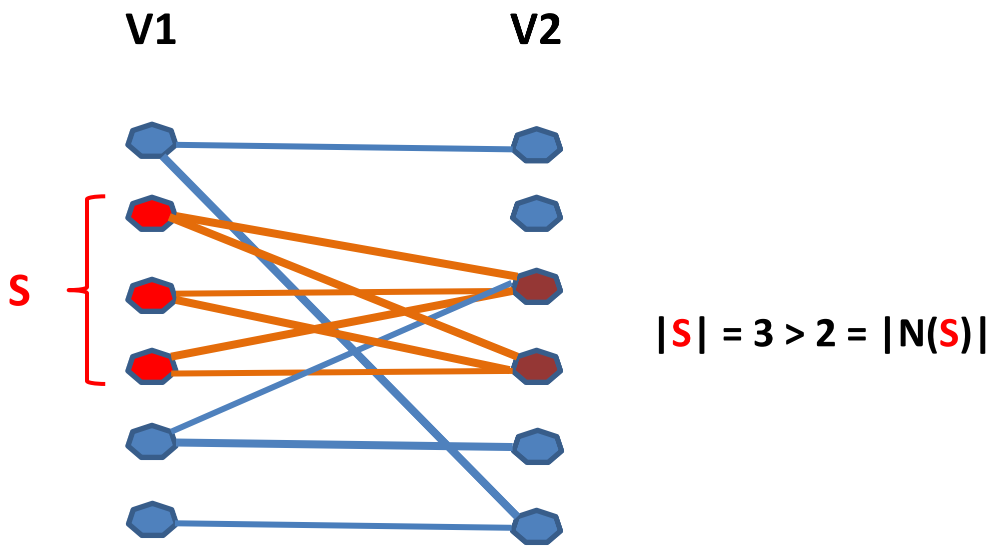
PERFECT MATCHING is in NP ∩ co-NP

Decision Problem PM: Given a **bipartite graph G(V1,V2; E)**, is there a **Perfect Matching M?**

- If **yes**, can exhibit a Perfect Matching **M** ⊆ **E**
- If **no**, can exhibit a set of nodes **S** such that **|N(S)| < |S|.**



Yes Instance G(V1,V2;E)
Certifier **M** ⊆ **E**



No Instance G(V1,V2;E)
Disqualifier **S** ⊆ **V1**

Good Characterizations

Observation. $P \subseteq NP \cap co\text{-}NP$.

Fundamental open question. Does $P = NP \cap co\text{-}NP$?

- Mixed opinions.
- Many examples where problem found to have a non-trivial good characterization, but only years later discovered to be in P.
  - linear programming [Khachiyan, 1979]
  - primality testing [Agrawal-Kayal-Saxena, 2002]

THM. Factoring is in $NP \cap co\text{-}NP$, but not known to be in P.

if poly-time algorithm for factoring,
can break RSA cryptosystem

↑

PRIMES is in NP ∩ co-NP

PRIMES = Given an odd integer $s > 0$; Is s PRIME ?

THM A. PRIMES is in NP ∩ co-NP.

Proof. We already know that PRIMES is in co-NP, so it suffices to prove that PRIMES is in NP.

Pratt's Theorem. An odd integer s is **prime** iff there exists an integer t **s.t.**

$1 < t < s$ s.t.

$$t^{s-1} \equiv 1 \pmod{s} \quad (a)$$
$$t^{(s-1)/p} \not\equiv 1 \pmod{s} \quad (b)$$

for all prime divisors p of s-1

Input. s = 437,677

Certificate. t = 17, $2^2 \times 3 \times 36473$

prime factorization of **s-1**
also need a **recursive** certificate
to assert that **3** and 36473 are **prime**

Certifier.

- Check s-1 = $2 \times 2 \times 3 \times 36{,}473$.
- Check $17^{s-1} = 1 \pmod{s}$ (a)
- Check $17^{(s-1)/2} \equiv 437{,}676 \pmod{s}$ (b)
- Check $17^{(s-1)/3} \equiv 329{,}415 \pmod{s}$ (b)
- Check $17^{(s-1)/36473} \equiv 305{,}452 \pmod{s}$ (b)

use repeated squaring

FACTOR is in NP ∩ co-NP

FACTORIZE (Search Problem). Given an integer **x**, find its **prime factorization**.

FACTOR (Decision Problem). Given two integers x and y,
Does **x** have a nontrivial **factor** less than **y**?

Theorem. **FACTOR** $\equiv_p$ FACTORIZE
Proof: Omitted (Binary Search and More Number Theory)

Theorem. **FACTOR** is in **NP ∩ co-NP**.
Proof.

- **Certificate**: a factor **p** of **x** that is less than **y**.

- **Disqualifier**: The prime factorization of **x** (where each prime factor is larger than y), along with a certificate that each factor is **prime (apply Pratt's Theorem and THM A in the previous slide)**.

Primality Testing and Factoring

We know: PRIMES $\leq_P$ FACTOR.

Natural question: Does FACTOR $\leq_P$ PRIMES ?
Consensus opinion. **No**.

State-of-the-art.
- **PRIMES** is in **P**. ← proved in 2001
- **FACTOR** **not** believed to be in **P**.

RSA cryptosystem.
- Based on dichotomy between complexity of two problems.
- To use **RSA**, must generate **large primes efficiently**.
- To break **RSA**, suffices to find efficient **factoring** algorithm.