CS 154

More NP-Complete Problems – and coNP

CS 154

Final Exam:

Friday March 18, 3:30-6:30pm

You're allowed one double-sided sheet of notes

Exam is comprehensive (but will emphasize computability / complexity topics)

Practice final will be released later this week

VOTE VOTE VOTE

For your favorite course on automata and complexity

Please complete
the online course evaluation

The Clique Problem

Given a graph G and positive k, does G contain a complete subgraph on k nodes?

CLIQUE = { (G,k) | G is an undirected graph with a k-clique }

Theorem (Karp): CLIQUE is NP-complete

$3SAT \leq_P CLIQUE$

We transform a 3-cnf formula ϕ into (G,k) such that

$\phi \in 3SAT \Leftrightarrow (G,k) \in CLIQUE$

Let $C_1, C_2, \ldots, C_m$ be clauses of ϕ. Assign k := m.
Make a graph G with m *groups* of 3 nodes each.

Group *i* corresponds to clause C_i of ϕ

Each node in group *i* is labeled with a literal of C_i

Put edges between all pairs of nodes in different groups,
except pairs of nodes with labels x_i and $\neg x_i$

Put no edges between nodes in the same group

When done putting in all the edges, *erase* the labels

Claim: $\phi \in$ 3SAT $\Leftrightarrow$ (G,m) $\in$ CLIQUE

Claim: If $\phi \in$ 3SAT then (G,m) $\in$ CLIQUE

Proof: Let A be a SAT assignment of ϕ.

For every clause C of ϕ, some literal in C is set true by A

For every clause C, let v_C be a vertex from group C of G, whose label is a literal that is set true by A

Claim: S = $\{v_C \mid C \in \phi\}$ is an m-clique in G. (note |S|=m)

Proof: Let $v_C, v_{C'}$ be in S. If $(v_C, v_{C'}) \notin E$…

Then v_C and $v_{C'}$ must label *inconsistent* literals, call them x and $\neg$x

But assignment A cannot satisfy both x and $\neg$x

Therefore $(v_C, v_{C'}) \in E$, for all $v_C, v_{C'} \in S$.

Hence S is an m-clique, and (G,m) $\in$ CLIQUE

Claim: $\phi \in$ 3SAT $\Leftrightarrow$ (G,m) $\in$ CLIQUE

Claim: If (G,m) $\in$ CLIQUE then $\phi \in$ 3SAT

Proof: Let S be an m-clique of G.

We'll construct a satisfying assignment A of ϕ.

Claim: S contains *exactly one node* from each group.

For each variable x of ϕ, make variable assignment:

A(x) := 1, if there is a vertex v ϵ S with label x

A(x) := 0, otherwise

For all i = 1,…,m, one vertex from group i is in S.

Therefore, for all i = 1,…,m

A satisfies at least one literal in the ith clause of ϕ

Therefore A is a satisfying assignment to ϕ

Independent Set

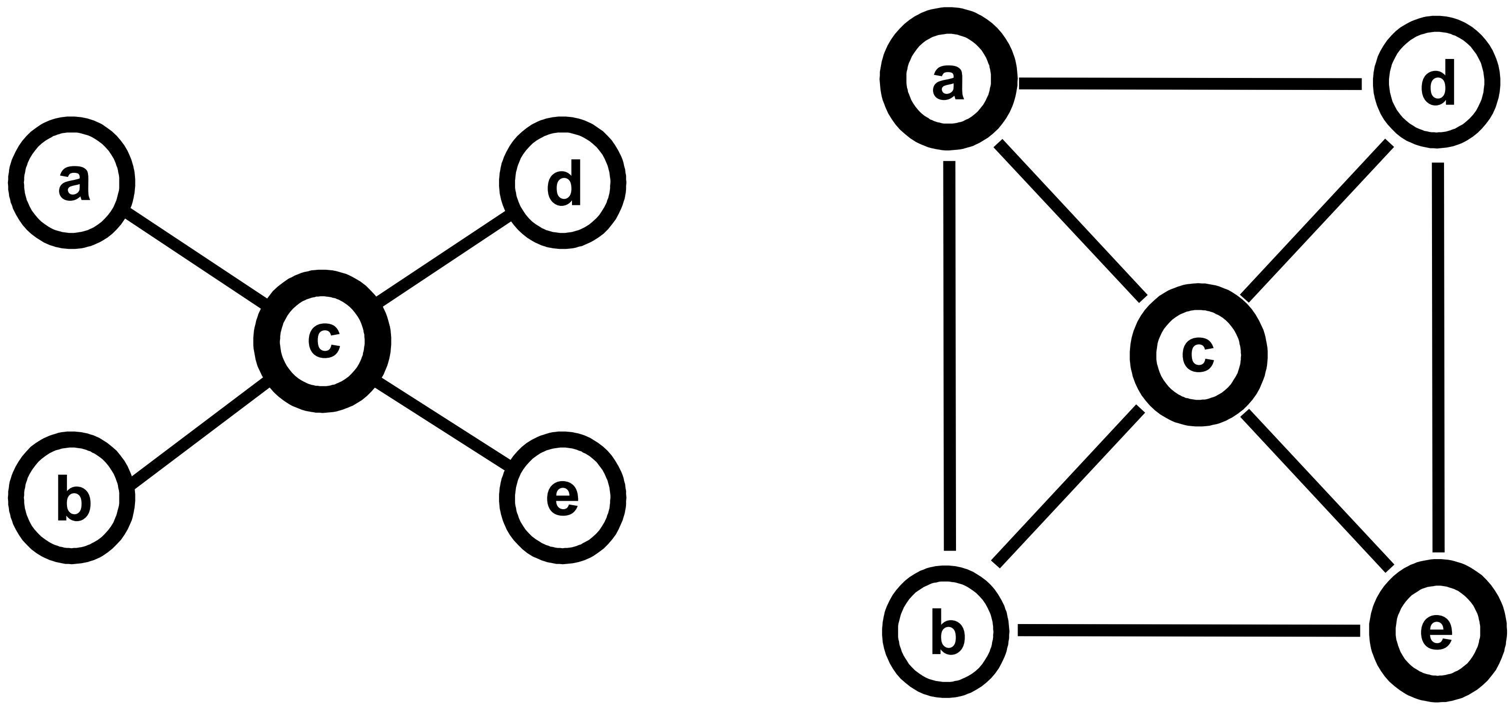
IS: Given a graph G = (V, E) and integer k,
is there $S \subseteq V$ such that $|S| \geq k$ and
***no* two vertices in S have an edge?**

CLIQUE: Given G = (V, E) and integer k,
is there $S \subseteq V$ such that $|S| \geq k$
and every pair of vertices in S have an edge?

CLIQUE $\leq_P$ IS:
Given G = (V, E), output G' = (V, E') where
$E' = \{(u,v) \mid (u,v) \notin E\}$.

$(G, k) \in$ CLIQUE iff $(G', k) \in$ IS

The Vertex Cover Problem



vertex cover = set of nodes C that cover all edges

For all edges, at least one endpoint is in C

VERTEX-COVER = { (G,k) | G is a graph with a vertex cover of size at most k}

Theorem: VERTEX-COVER is NP-Complete

(1) VERTEX-COVER $\in$ NP

(2) IS $\leq_P$ VERTEX-COVER

IS $\leq_P$ VERTEX-COVER

Want to transform a graph G and integer k into G' and k' such that

$$(G,k) \in IS \Leftrightarrow (G',k') \in \text{VERTEX-COVER}$$

IS $\leq_P$ VERTEX-COVER

Claim: For every graph $G = (V,E)$, and subset $S \subseteq V$,

S is an independent set

if and only if $(V - S)$ is a vertex cover

Proof: S is an independent set

$\Leftrightarrow$ $(\forall\, u, v \in V)[\, (u \in S \text{ and } v \in S) \Rightarrow (u,v) \notin E\,]$

$\Leftrightarrow$ $(\forall\, u, v \in V)[\, (u,v) \in E \Rightarrow (u \notin S \text{ or } v \notin S)\,]$

$\Leftrightarrow$ $(V - S)$ is a vertex cover

Therefore $(G,k) \in IS \Leftrightarrow (G,|V| - k) \in$ VERTEX-COVER

Our polynomial time reduction: $f(G,k) := (G, |V| - k)$

The Subset Sum Problem

Given: Set $S = \{a_1, \ldots, a_n\}$ of positive integers and a positive integer t

Is there an $A \subseteq \{1, \ldots, n\}$ such that $t = \sum_{i \in A} a_i$?

SUBSET-SUM = $\{(S, t) \mid \exists\, S' \subseteq S$ s.t. $t = \sum_{b \in S'} b\}$

A simple number-theoretic problem!

Theorem: SUBSET-SUM is NP-complete

VC $\leq_P$ SUBSET-SUM

Want to reduce a *graph* to a *set of numbers*

Given (G, k), let $E = \{e_0, \ldots, e_{m-1}\}$ and $V = \{1, \ldots, n\}$

Our subset sum instance (S, t) will have $|S| = n+m$

"Edge numbers":

For every $e_j \in E$, put $b_j = 4^j$ in S

"Node numbers":

For every $i \in V$, put $a_i = 4^m + \sum_{j\,:\,i \in e_j} 4^j$ in S

Set the target number: $t = k \cdot 4^m + \sum_{j=0}^{m-1} (2 \cdot 4^j)$

For every $e_j \in E$, put $b_j = 4^j$ in S

For every $i \in V$, put $a_i = 4^m + \sum_{j\,:\,i \in e_j} 4^j$ in S

Set $t = k \cdot 4^m + \sum_{j=0}^{m-1} (2 \cdot 4^j)$

Claim: If $(G,k) \in VC$ then $(S,t) \in$ SUBSET-SUM

Suppose $C \subseteq V$ is a VC with k vertices.

Let $S' = \{a_i : i \in C\} \cup \{b_j : |e_j \cap C| = 1\}$

S' = (*node numbers* corresponding to nodes in C) *plus*
(*edge numbers* corresponding to edges covered *only once* by C)

Claim: The sum of all numbers in S' equals t!

Think of the numbers as being in "base 4"…
as vectors with m+1 components

For every $e_j \in E$, put $b_j = 4^j$ in S

For every $i \in V$, put $a_i = 4^m + \sum_{j\,:\,i \in e_j} 4^j$ in S

Set $t = k \cdot 4^m + \sum_{j=0}^{m-1} (2 \cdot 4^j)$

Claim: If (S,t) $\in$ SUBSET-SUM then (G,k) $\in$ VC

Suppose $C \subseteq V$ and $F \subseteq E$ satisfy

$$\sum_{i \in C} a_i + \sum_{e_j \in F} b_j = t = k \cdot 4^m + \sum_{j=0}^{m-1} (2 \cdot 4^j)$$

Claim: C is a vertex cover of size k.

Proof: Subtract out the b_j numbers from the above sum.

What remains is a sum of the form:

$$\sum_{i \in C} a_i = k \cdot 4^m + \sum_{j=0}^{m-1} (c_j \cdot 4^j)$$

where each $c_j > 0$. But c_j = number of nodes in C covering e_j

This implies C is a vertex cover!

The Knapsack Problem

Given: $S = \{(v_1,c_1)\ldots, (v_n,c_n)\}$ of pairs of positive integers
a capacity budget C
a value V

Is there an $S' \subseteq \{1,\ldots,n\}$ such that
$(\sum_{i \in S'} v_i) \geq V$ and $(\sum_{i \in S'} c_i) \leq C$?

Define KNAPSACK = {(S, C, V) | the answer is yes}

A classic economics/logistics problem!

Theorem: KNAPSACK is NP-complete

KNAPSACK is NP-complete

KNAPSACK is in NP?

Theorem: SUBSET-SUM $\leq_P$ KNAPSACK

Proof: Given an instance ($S = \{a_1,\ldots,a_n\}$, t)
of SUBSET-SUM, create a KNAPSACK instance:

For all i, set $(p_i, c_i) := (a_i, a_i)$

Define $T = \{(p_1, c_1),\ldots, (p_n, c_n)\}$

Define $C := P := t$

Then, $(S,t) \in$ SUBSET-SUM $\Leftrightarrow$ $(T,C,P) \in$ KNAPSACK

Subset of S that sums to t =
Solution to the Knapsack instance!

The Partition Problem

Given: Set $S = \{a_1, \ldots, a_n\}$ of positive integers

Is there an $S' \subseteq S$ such that $(\sum_{a\_i \in S'} a_i) = (\sum_{a\_i \in S-S'} a_i)$?

(Formally, PARTITION is the set of all S such that the answer to this question is yes.)

In other words, is there a way to partition S into two parts, with equal sum in both parts?

A problem in fair division

Theorem: PARTITION is NP-complete

PARTITION is NP-complete

(1) PARTITION is in NP

(2) SUBSET-SUM $\leq_P$ PARTITION

Given: Set $S = \{a_1, \ldots, a_n\}$ of positive integers
positive integer t

Output $T := \{a_1, \ldots, a_n, 2A-t, A+t\}$, where $A := \sum_i a_i$

Claim: $(S,t) \in$ SUBSET-SUM $\Leftrightarrow T \in$ PARTITION

Given: Set $S = \{a_1, \ldots, a_n\}$ of positive integers
positive integer t

Output $T := \{a_1, \ldots, a_n, 2A-t, A+t\}$, where $A := \sum_i a_i$

Claim: $(S,t) \in$ SUBSET-SUM $\Leftrightarrow T \in$ PARTITION

What's the sum of all numbers in T? 4A

Therefore: $T \in$ PARTITION
$\Leftrightarrow$ There is a $T' \subseteq T$ that sums to 2A.

Proof of: $(S,t) \in$ SUBSET-SUM $\Rightarrow T \in$ PARTITION:

If $(S,t) \in$ SUBSET-SUM, let $S' \subseteq S$ sum to t.
Then $S' \cup \{2A-t\} \subseteq T$ sums to 2A, so $T \in$ PARTITION

Given: Set $S = \{a_1,\ldots, a_n\}$ of positive integers
positive integer t

Output $T := \{a_1,\ldots, a_n, 2A\text{-}t, A+t\}$, where $A := \sum_i a_i$

Claim: $(S,t) \in$ SUBSET-SUM $\Leftrightarrow T \in$ PARTITION

$T \in$ PARTITION $\Leftrightarrow$ There is a $T' \subseteq T$ that sums to 2A.

Proof of: $T \in$ PARTITION $\Rightarrow (S,t) \in$ SUBSET-SUM

If $T \in$ PARTITION, let $T' \subseteq T$ be a subset that sums to 2A.

Observation: Exactly *one* of $\{2A\text{-}t, A+t\}$ is in T'.

If $(2A\text{-}t) \in T'$, then $T' - \{2A\text{-}t\}$ sums to t.

But $T' - \{2A\text{-}t\}$ is a subset of S! So $(S,t) \in$ SUBSET-SUM

If $(A+t) \in T'$, then $(T - T') - \{2A\text{-}t\}$ sums to $(2A - (2A\text{-}t)) = t$

Note that $(T - T') - \{2A\text{-}t\}$ is a subset of S.
Therefore $(S,t) \in$ SUBSET-SUM

The Bin Packing Problem

Given: Set S = $\{a_1, \ldots, a_n\}$ of positive integers,
a bin capacity B, and a target integer K.
Can we partition S into K subsets such that
each subset sums to at most B?

Is there a way to pack the items of S into K bins,
with each bin having capacity B?

Ubiquitous in shipping and optimization

Theorem: BIN PACKING is NP-complete

BIN PACKING is NP-complete

BIN PACKING is in NP?

Theorem: PARTITION $\leq_P$ BIN PACKING

Proof: Given an instance $S = \{a_1, \ldots, a_n\}$ of PARTITION, create an instance of BIN PACKING with:

$$S = \{a_1, \ldots, a_n\}$$
$$B = (\textstyle\sum_i a_i)/2$$
$$k = 2$$

Then, $S \in$ PARTITION $\Leftrightarrow$ $(S,B,k) \in$ BIN PACKING:

Partition of S into two equal sums = Solution to the Bin Packing instance!

Two Problems

Let G denote a graph, and s and t denote nodes.

SHORTEST PATH
= {(G, s, t, k) |
G has a simple path of length < k from s to t }

LONGEST PATH
= {(G, s, t, k) |
G has a simple path of length > k from s to t }

Are either of these in P? Are both of them?

HAMPATH = { (G,s,t) | G is an directed graph with a Hamiltonian path from s to t}

Theorem: HAMPATH is NP-Complete

(1) HAMPATH $\in$ NP

(2) 3SAT $\leq_P$ HAMPATH

See Sipser for the proof

HAMPATH $\leq_P$ LONGEST-PATH

LONGEST-PATH
= {(G, s, t, k) |
G has a simple path of length > k from s to t }

Can reduce HAMPATH to LONGEST-PATH by observing:

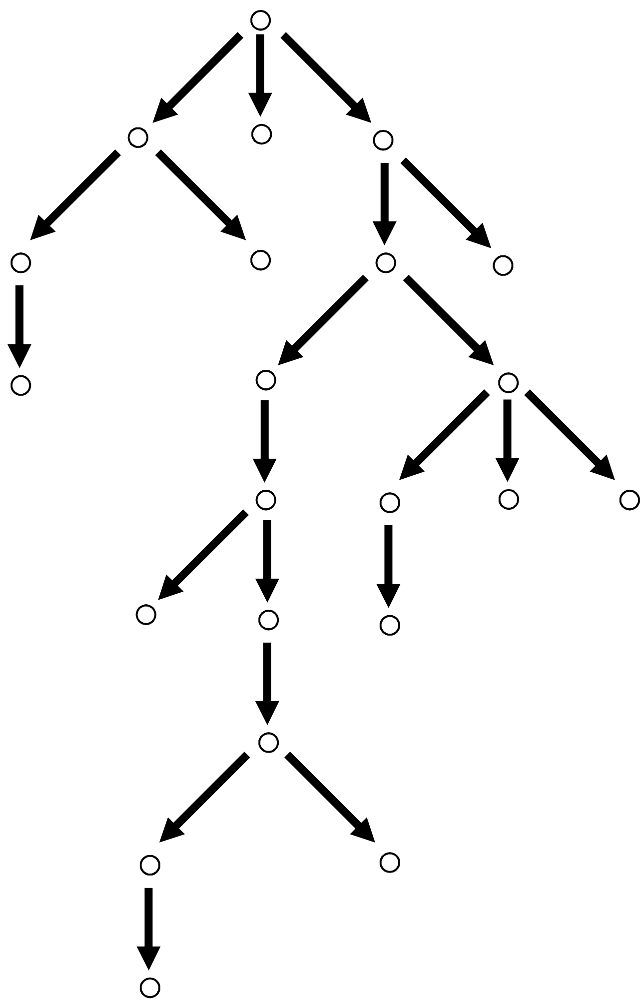
(G, s, t) $\in$ HAMPATH
$\Leftrightarrow$ (G, s, t, |V|) $\in$ LONGEST-PATH

Therefore LONGEST-PATH is NP-hard.

coNP and Friends

Definition: coNP = { L | $\neg$L $\in$ NP }

What does a coNP computation look like?

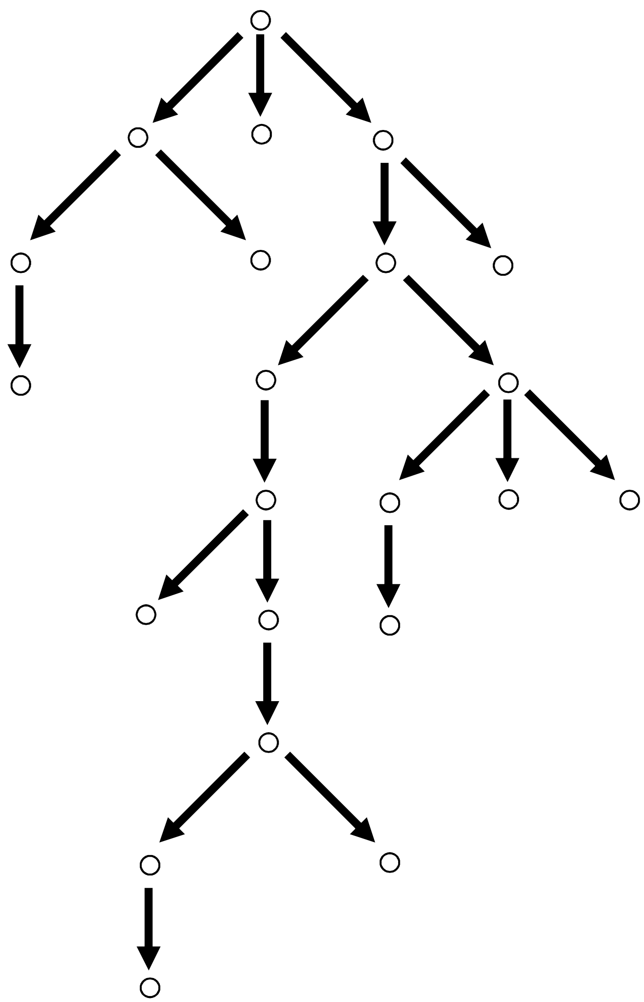


A *co-nondeterministic* machine has multiple computation paths, and has the following behavior:

- **the machine accepts if *all paths reach* accept state**
- **the machine rejects if *at least one path* reaches reject state**

Definition: coNP = { L | $\neg$L $\in$ NP }

What does a coNP computation look like?



In NP algorithms, we can use a "guess" instruction in pseudocode:
Guess string y of |x|^k length…
and the machine accepts if some y leads to an accept state

In coNP algorithms, we can use a "try all" instruction:
Try all strings y of |x|^k length…
and the machine accepts if every y leads to an accept state

TAUTOLOGY = { ϕ | ϕ is a Boolean formula and every variable assignment satisfies ϕ }

Theorem: TAUTOLOGY is in coNP

How would we write pseudocode for a coNP machine that decides TAUTOLOGY?

How would we write TAUTOLOGY as the complement of some NP language?

Is P $\subseteq$ coNP?

Yes!

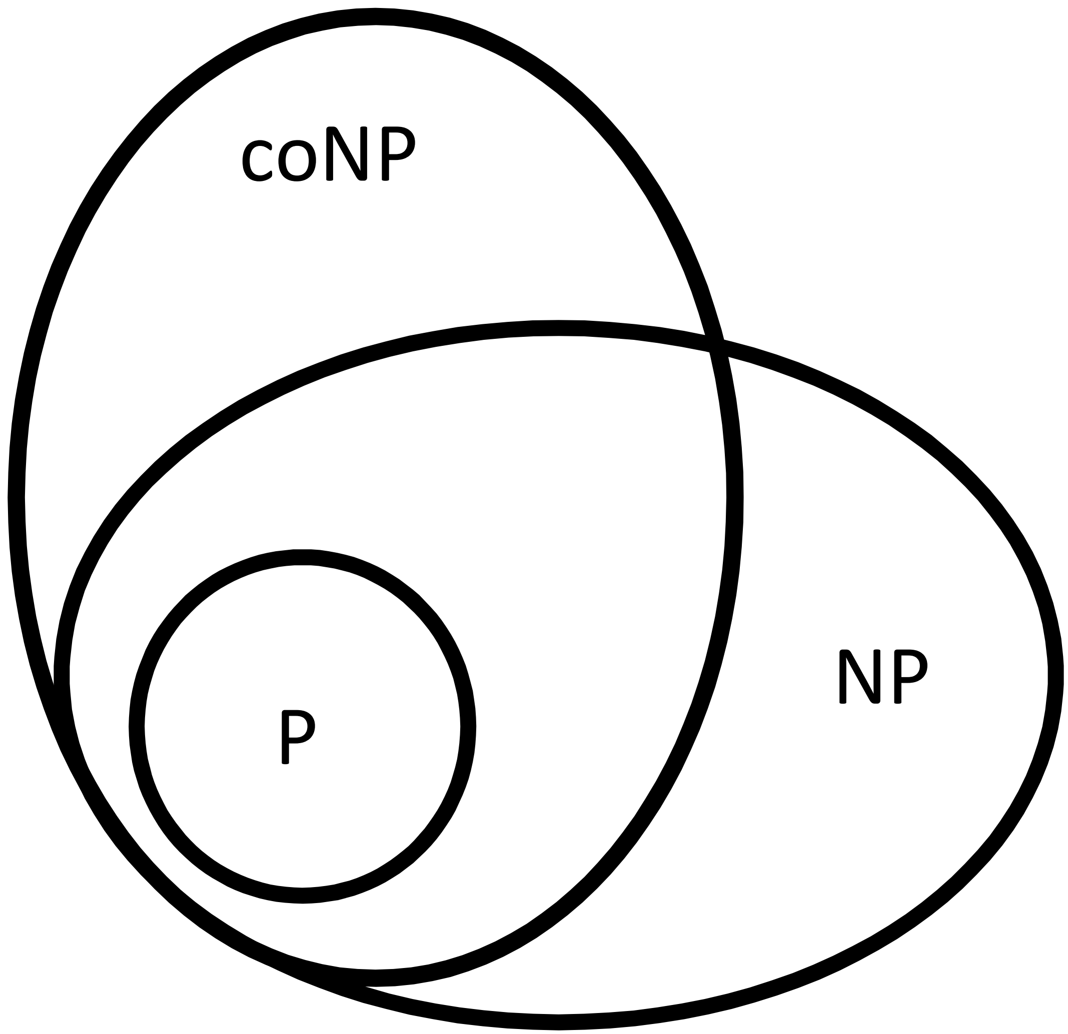
$L \in P$ implies that $\neg L \in P$
(hence $\neg L \in NP$)

In general, *deterministic* complexity classes are closed under complement

Is NP = coNP?

THIS IS AN OPEN QUESTION!

It is believed that NP $\neq$ coNP


coNP
NP
P