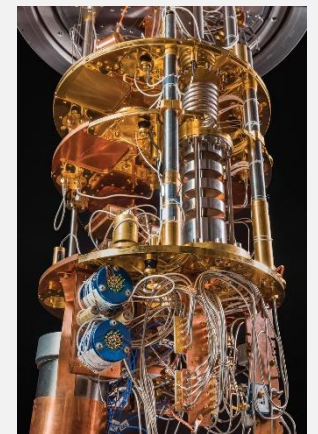
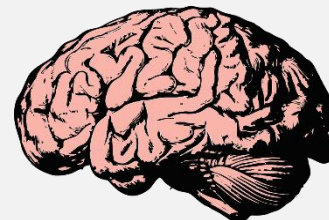
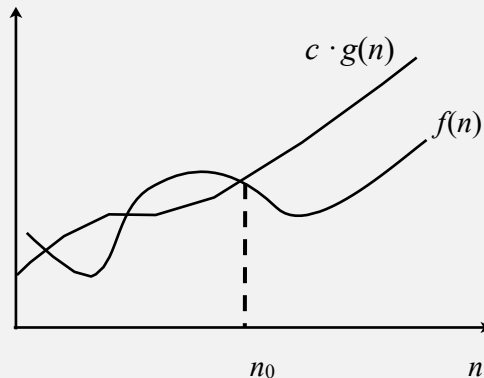
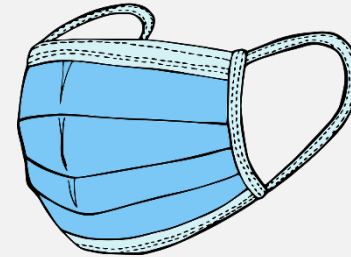
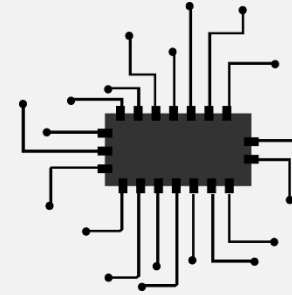
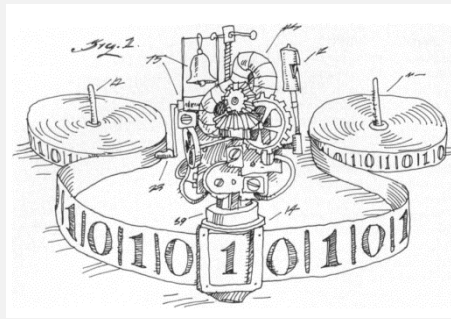




COMP 3602

Intro to Theory of Computation

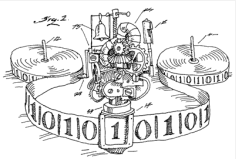
Lecture 16

Announcements



- Computability drill due Mar 12
 - One more drill (on complexity) after that
- Drop date was extended till March 10th
- Two more tests left:
 - Oral exam on NP-completeness *Fv1 20+η*
 - Written exam on all the material we covered in the course.

Questions/comments?



Scaling down to complexity

“efficient”, “small”, “easy” $\equiv$ polynomial

- Decidable
 - Turing-recognizable
- For some decidable R
- $$L = \{x \mid \exists y R(x, y)\}$$

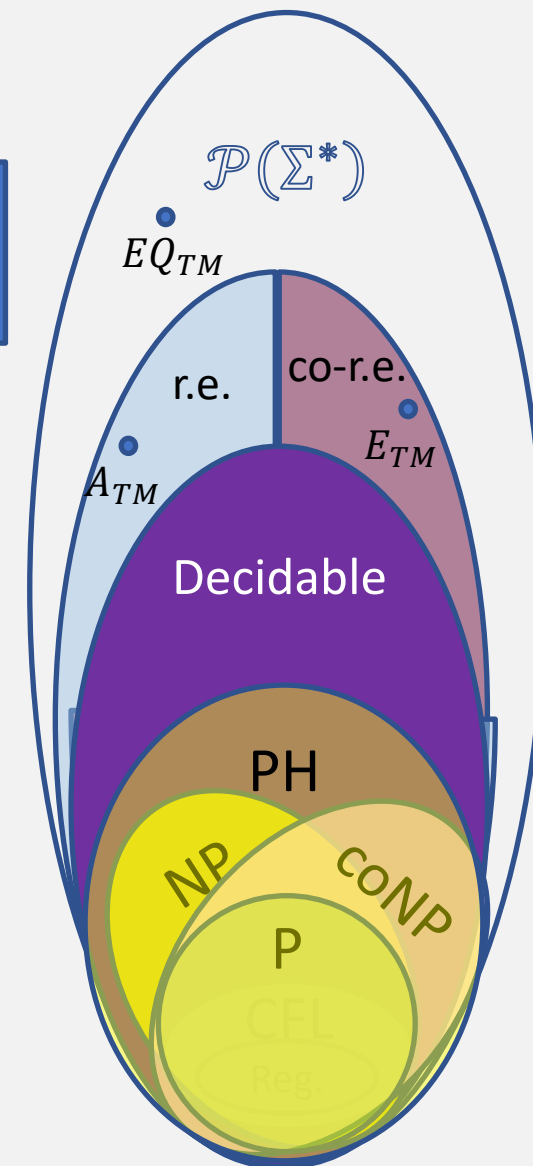
– P

– NP

For some **polytime** decidable R ,
some constants c, d

$$L = \{x \mid \exists y \mid y \mid \leq cn^d \wedge R(x, y)\}$$

- Co-r.e
- Arithmetic hierarchy
- $CoNP$
- Polytime hierarchy (PH)

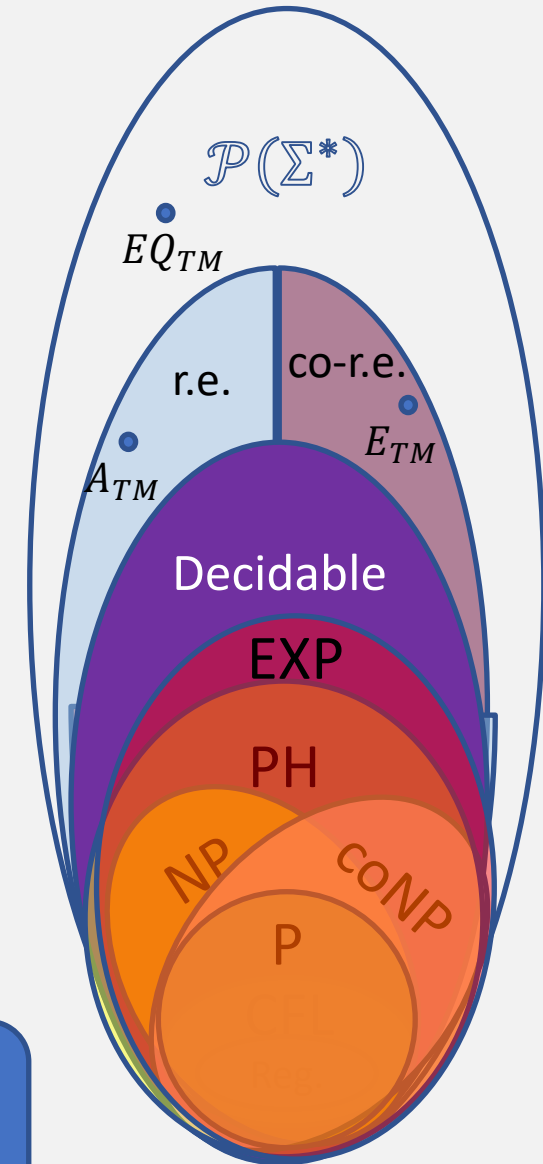


Complexity class NP

Complexity class NP: Problems for which *checking* solutions is in polynomial time: $L = \{x \mid \exists y \ |y| \leq cn^d \wedge R(x, y)\}$

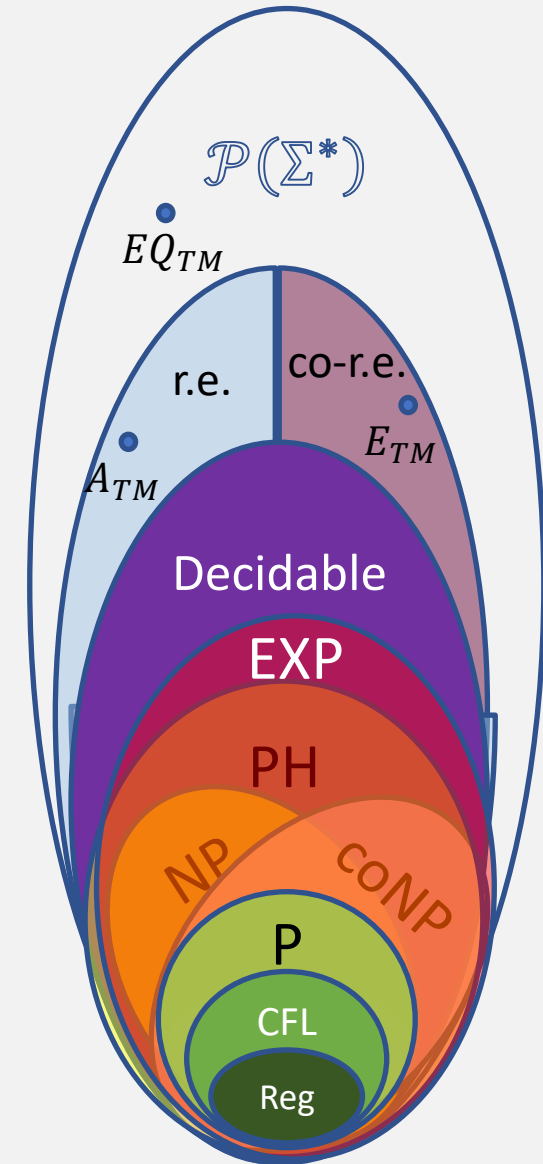
- Theorem: $P \subseteq NP \subseteq EXP$
 - $P \subseteq NP$: if an algorithm found solution, it exists. To check, compute it.
 - $NP \subseteq EXP$: There are $2 \cdot 2^{cn^d} - 1$ possible solutions y of size $\leq cn^d$ and checking each one takes polynomial time (say $O(n^k)$). Checking all takes time $O(n^k \cdot 2^{cn^d})$
- Note: NP contains a lot of easy problems! Exponential time is only an upper bound (O rather than Θ)

Are there any problems in NP for which the brute-force algorithm above is the fastest? That's a million dollar question!



Time complexity classes

- $T_M(n) = \max_{x, |x|=n} \{ \text{number of steps } M \text{ takes to halt on } x \}$
- $DTime(f(n)) = \{L \mid \exists M \ \mathcal{L}(M) = L \text{ and } T_M \in O(f)\}$
 - Where M is a deterministic TM.
- $P = \bigcup_{d \in \mathbb{N}} DTime(n^d)$
- $EXP = \bigcup_{d \in \mathbb{N}} DTime(2^{n^d})$
- $L \in NP$ iff for some **polytime** decidable R , **some constants** c, d , $L = \{x \mid \exists y \ |y| \leq cn^d \wedge R(x, y)\}$
- $L \in coNP$ iff $\bar{L} \in NP$
- k -th level of PH: generalizes NP and coNP to k quantifier alternations, all variables bounded by cn^d .
 - k is constant.
 - Σ_k^p is when the k -quantifier condition starts with $\exists$, Π_k^p with $\forall$.

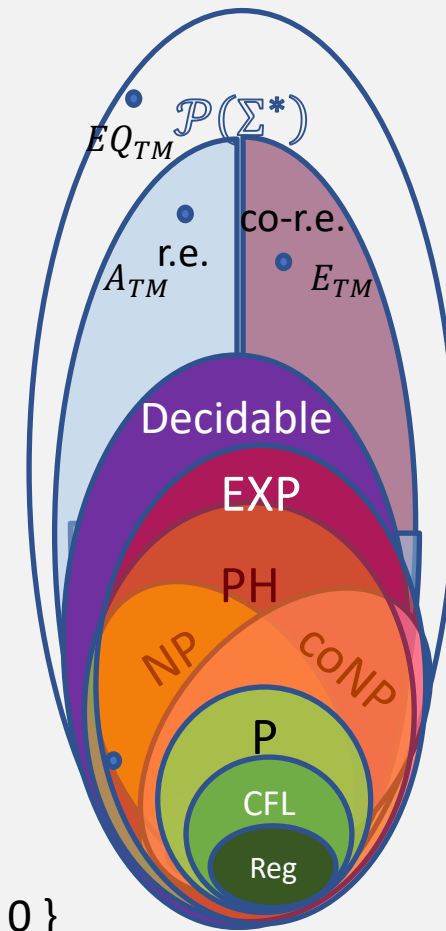


In P

- Graph completeness:
 - $\{\langle G \rangle \mid G \text{ is a complete graph}\}$
- Shortest path:
 - $\{\langle G, s, t, k \rangle \mid G \text{ has path from } s \text{ to } t \text{ of length } \leq k\}$
- 2SAT (2CNF-satisfiability):
 - $\{\langle \varphi \rangle \mid \varphi \text{ is a satisfiable 2-CNF formula}\}$
- Circuit evaluation
 - $\{\langle C, x \rangle \mid C \text{ is a Boolean circuit such that } C(x) = 1\}$
- Linear equation solvability
 - $\{\langle S \rangle \mid S \text{ is a solvable system of linear equations}\}$
- Primality
 - $\{\langle k \rangle \mid k \text{ is a prime number}\}$

In NP, not known to be in P

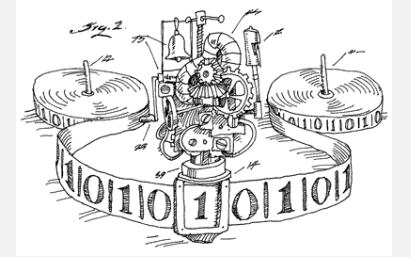
- CLIQUE:
 - $\{\langle G, k \rangle \mid \text{graph } G \text{ has a complete subgraph on } k \text{ vertices}\}$
- Longest path:
 - $\{\langle G, k \rangle \mid G \text{ has a simple path containing at least } k \text{ vertices}\}$
- 3SAT (3CNF-satisfiability):
 - $\{\langle \varphi \rangle \mid \varphi \text{ is a satisfiable 3-CNF formula}\}$
- CircuitSAT (circuit satisfiability):
 - $\{\langle C \rangle \mid C \text{ is a Boolean circuit computing } f \not\equiv 0\}$
- Quadratic equation solvability
 - $\{\langle S \rangle \mid S \text{ is a solvable system of quadratic equations}\}$



Factoring is not a language, so it is not in NP (though it is efficiently verifiable)

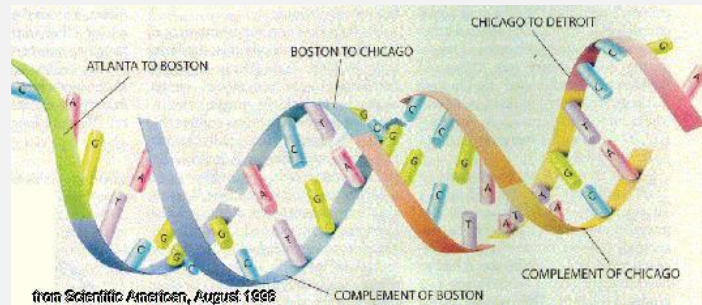
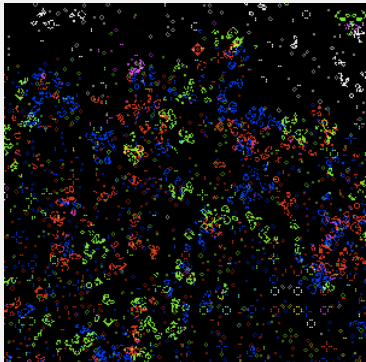


How to compare *decidable* languages?



Reductions

Scaling down “computable” to “polytime computable” gives polytime reductions





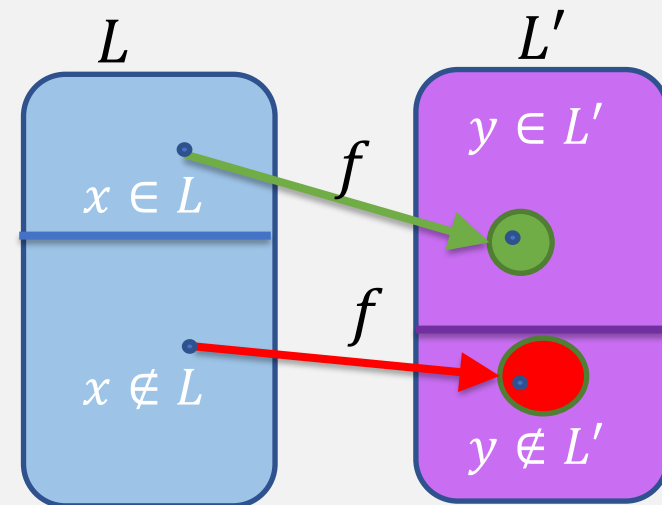
Polytime reductions

- Recall: L is **mapping-reducible** to L' , written $L \leq_m L'$ iff there exists a computable function $f: \Sigma^* \rightarrow \Sigma^*$ such that $\forall x \in \Sigma^*, x \in L \Leftrightarrow f(x) \in L'$
 - But computable functions can just solve NP, EXP and so on problems!
 - Need a weaker reduction than the languages we want to compare.

Definition: L is **polytime mapping-reducible** to L' , written $L \leq_p L'$ iff there exists a **polytime** computable function $f: \Sigma^* \rightarrow \Sigma^*$ such that

$$\forall x \in \Sigma^*, x \in L \Leftrightarrow f(x) \in L'$$

- If $L \leq_p L'$, then L is at most as hard as L'
 - $\leq_p$ is “hardness greater than”
 - If $L \leq_p L'$, and L is “hard”, then L' must be “hard”
- If $L_1 \leq_p L_2$ and $L_2 \leq_p L_3$ then $L_1 \leq_p L_3$
- If $L_1 \leq_p L_2$, then $\overline{L_1} \leq_p \overline{L_2}$





Clique and Independent set

- Let $G = (V, E)$ be an undirected graph.
- A “clique” is a complete subgraph of G
 - $S \subseteq V$ and $\forall u \in V \forall v \in V (u \in S \wedge v \in S \rightarrow E(u, v))$
- An “independent set” is a set of vertices with no edges between any pair
 - $S \subseteq V$ and $\forall u \in V \forall v \in V (u \in S \wedge v \in S \rightarrow \neg E(u, v))$
- CLIQUE: $\{\langle G, k \rangle \mid \text{graph } G \text{ has a clique } S \text{ st } |S| \geq k\}$
 - $\{\langle G, k \rangle \mid \exists S \subseteq V |S| \geq k \wedge \forall u \in S \forall v \in S E(u, v)\}$
- IndSet: $\{\langle G, k \rangle \mid \text{graph } G \text{ has an independent set } S \text{ st } |S| \geq k\}$
 - $\{\langle G, k \rangle \mid \exists S \subseteq V |S| \geq k \wedge \forall u \in S \forall v \in S \neg E(u, v)\}$
- Both CLIQUE and IndSet are in NP
 - Can check that S is a clique (independent set) in time $O(|S|^2)$
 - If have random access to E , otherwise at most $O(|E|^2 |S|^2)$



$\text{IndSet} \leq_p \text{CLIQUE}$

- **CLIQUE**: $\{\langle G, k \rangle \mid \text{graph } G \text{ has a clique } S \text{ such that } |S| \geq k\}$
 - $\{\langle G, k \rangle \mid \exists S \subseteq V \ |S| \geq k \wedge \forall u \in S \forall v \in S \ E(u, v)\}$
- **IndSet**: $\{\langle G, k \rangle \mid \text{graph } G \text{ has an independent set } S \text{ st } |S| \geq k\}$
 - $\{\langle G, k \rangle \mid \exists S \subseteq V \ |S| \geq k \wedge \forall u \in S \forall v \in S \ \neg E(u, v)\}$
- **Claim**: $\text{IndSet} \leq_p \text{CLIQUE}$
 - Construct a polytime computable function f
$$f(x) = \begin{cases} \langle G', k' \rangle & \text{if } x = \langle G, k \rangle \\ x & \text{otherwise} \end{cases}$$
 - Where $k' = k$ and $G' = (V, \bar{E})$
 - That is, G' is a complement of G .

We will often ignore the “garbage” case and just say what f does on correctly formed instances

- Write eg $f(\langle G, k \rangle) = \langle G', k' \rangle$ and say what $\langle G', k' \rangle$ is in terms of $\langle G, k \rangle$

