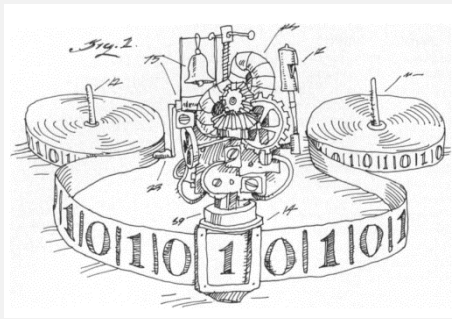
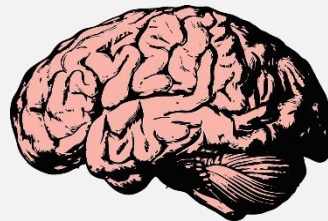
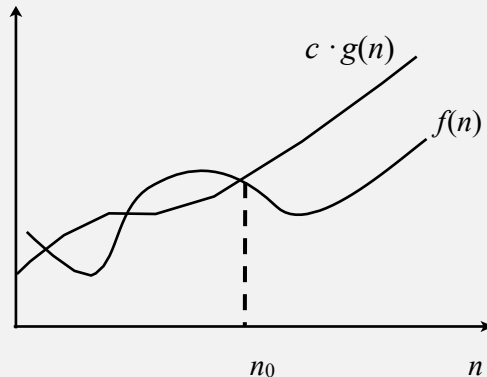


COMP 3602

Intro to Theory of Computation

Lecture 23

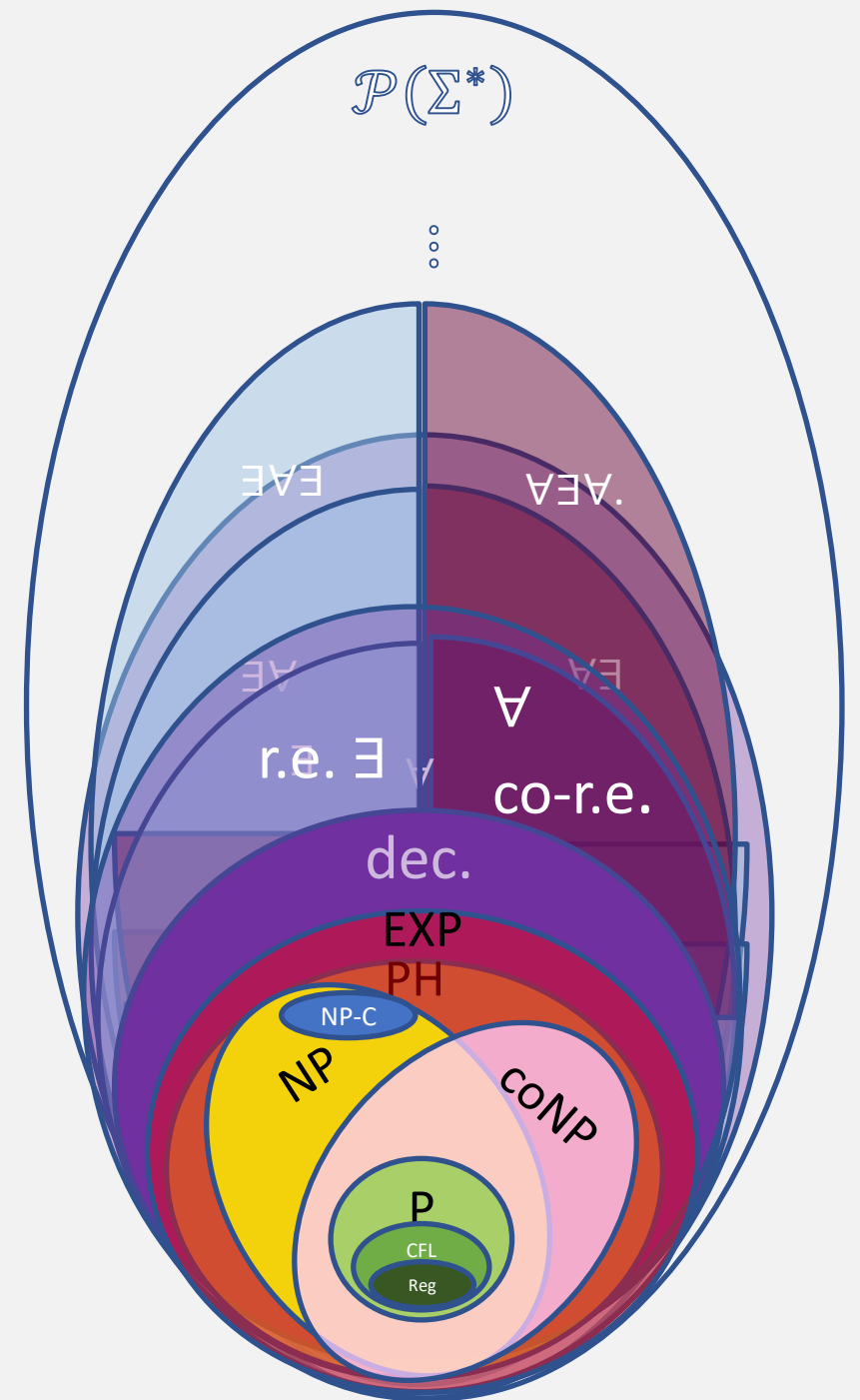


Announcements



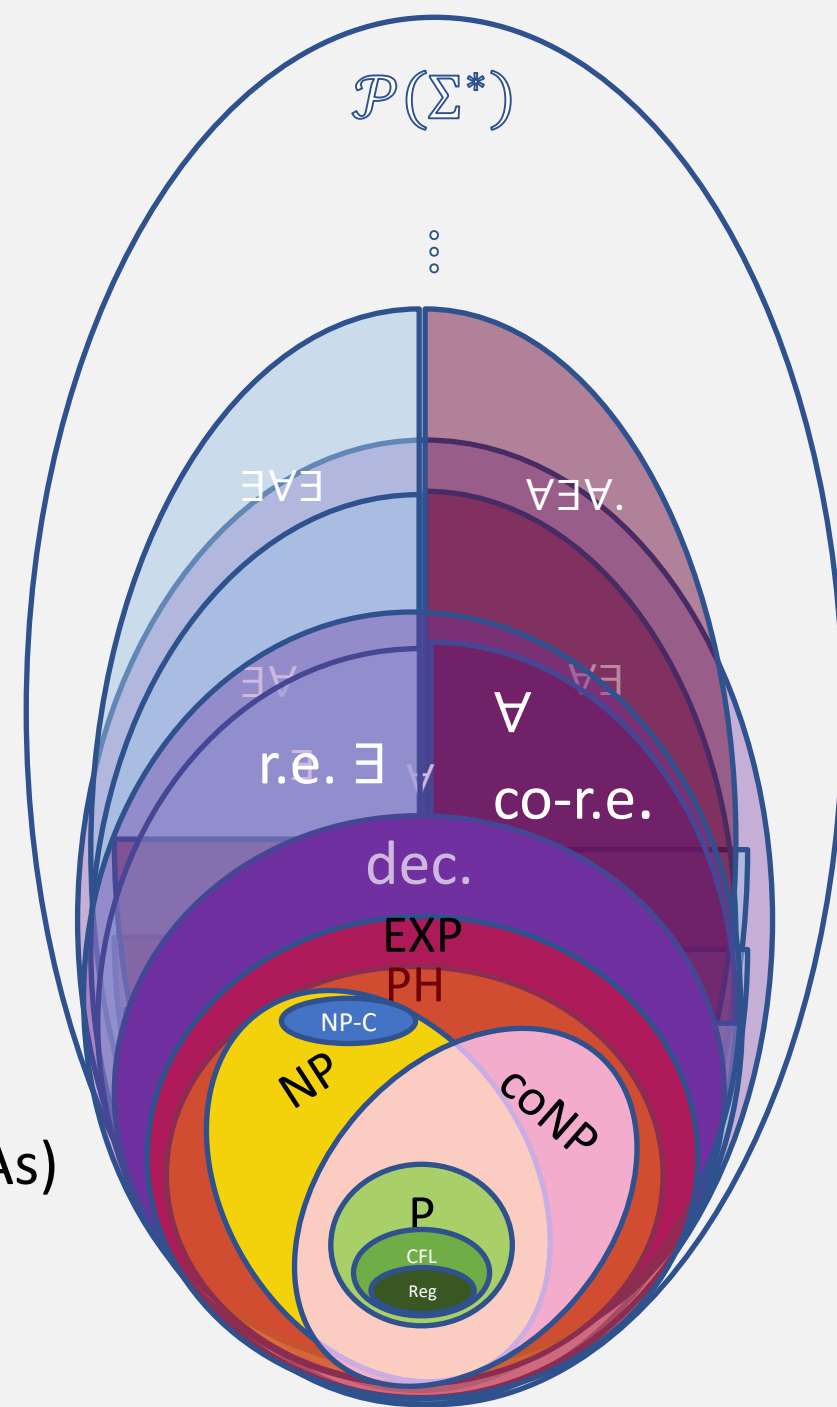
- All drills should be done by now
 - If you didn't get a 100, you still can redo them until next Wednesday
- Fri Mar 27th : Written exam on all the material we covered in the course.
 - In class, during class time
- Sun Mar 29: presentation slides are due on Brightspace
 - Presentations Mar 30 and Apr 1, please sign up for your preferred day
- Apr 1 is the last day of classes, no classes next Friday or the week after.

Questions/comments?



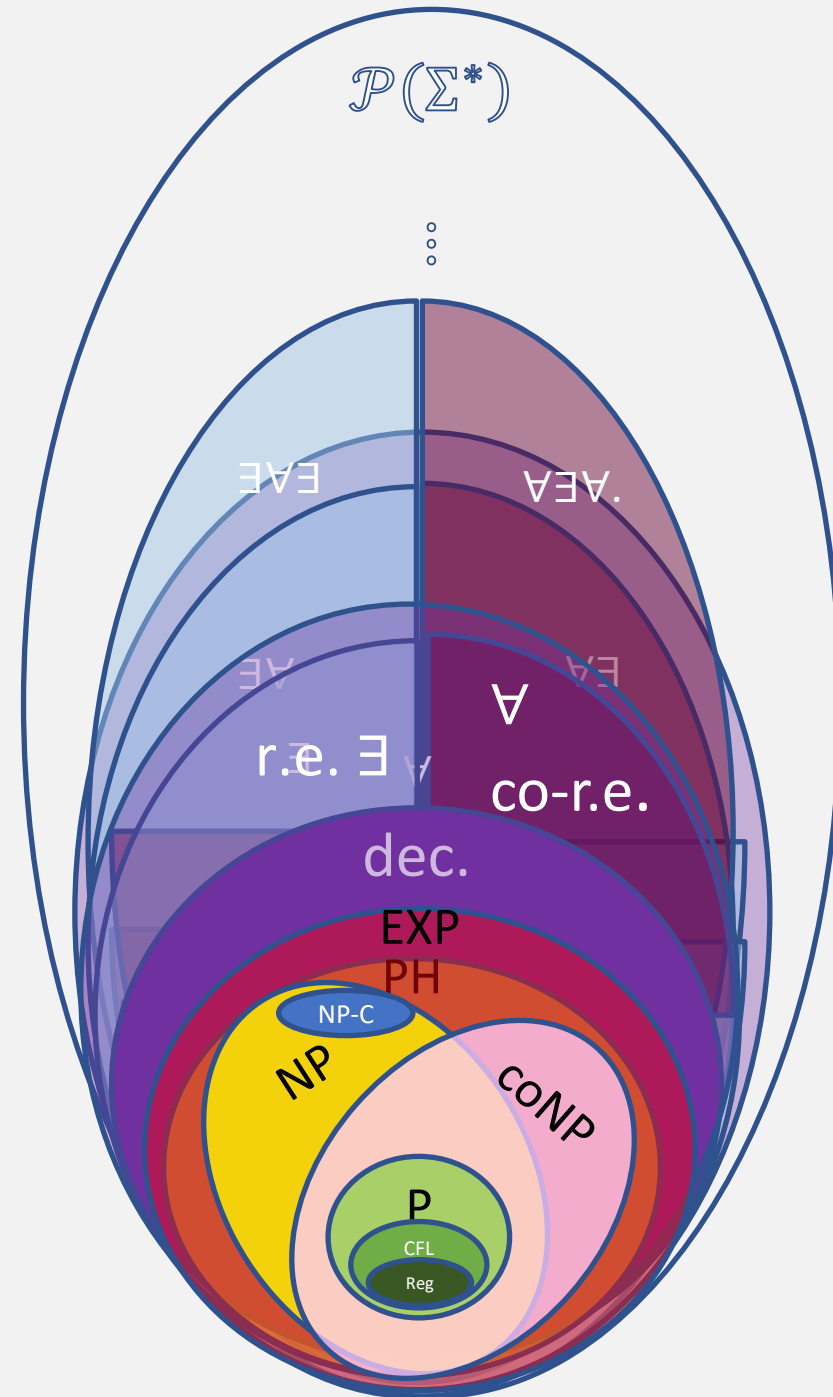
First topic: hardness and models

- Computational hardness
 - Representative problems: Is k even? prime? Its factors? Description?
 - Review of $O, \Omega, \Theta, o, \omega, \mathcal{O}$
 - Upper and lower bounds for comparison-based sorting
- Models of computation
 - Turing machines (and their variants)
 - Regular languages (exactly the class computed by DFAs)
 - Context-free languages (exactly the class computed by PDAs)



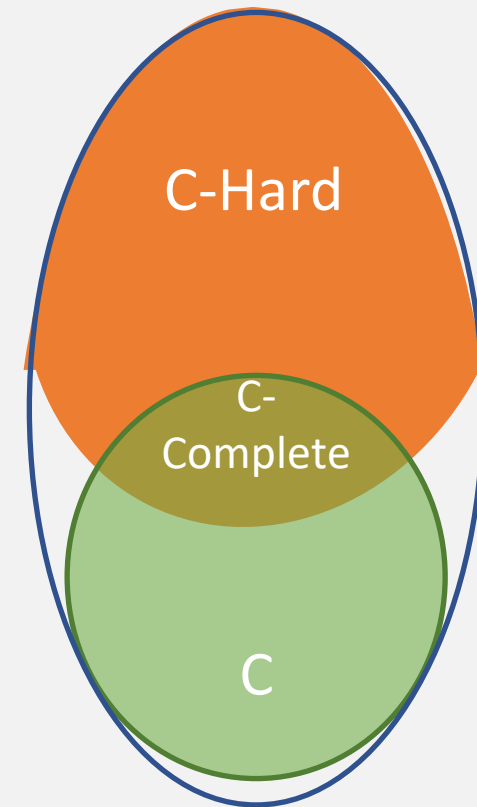
Models of computation

- Turing machines (of several kinds), DFAs, NFAs, PDAs...
- Simulation: model A simulates model B if, given any device of type B, can construct a device of type A accepting the same language.
 - Eg, given a DFA can construct an equivalent Turing machine.
- Equivalence: two models A and B are equivalent if A simulates B and B simulates A.
 - NFAs and DFAs are equivalent (though not efficiently)



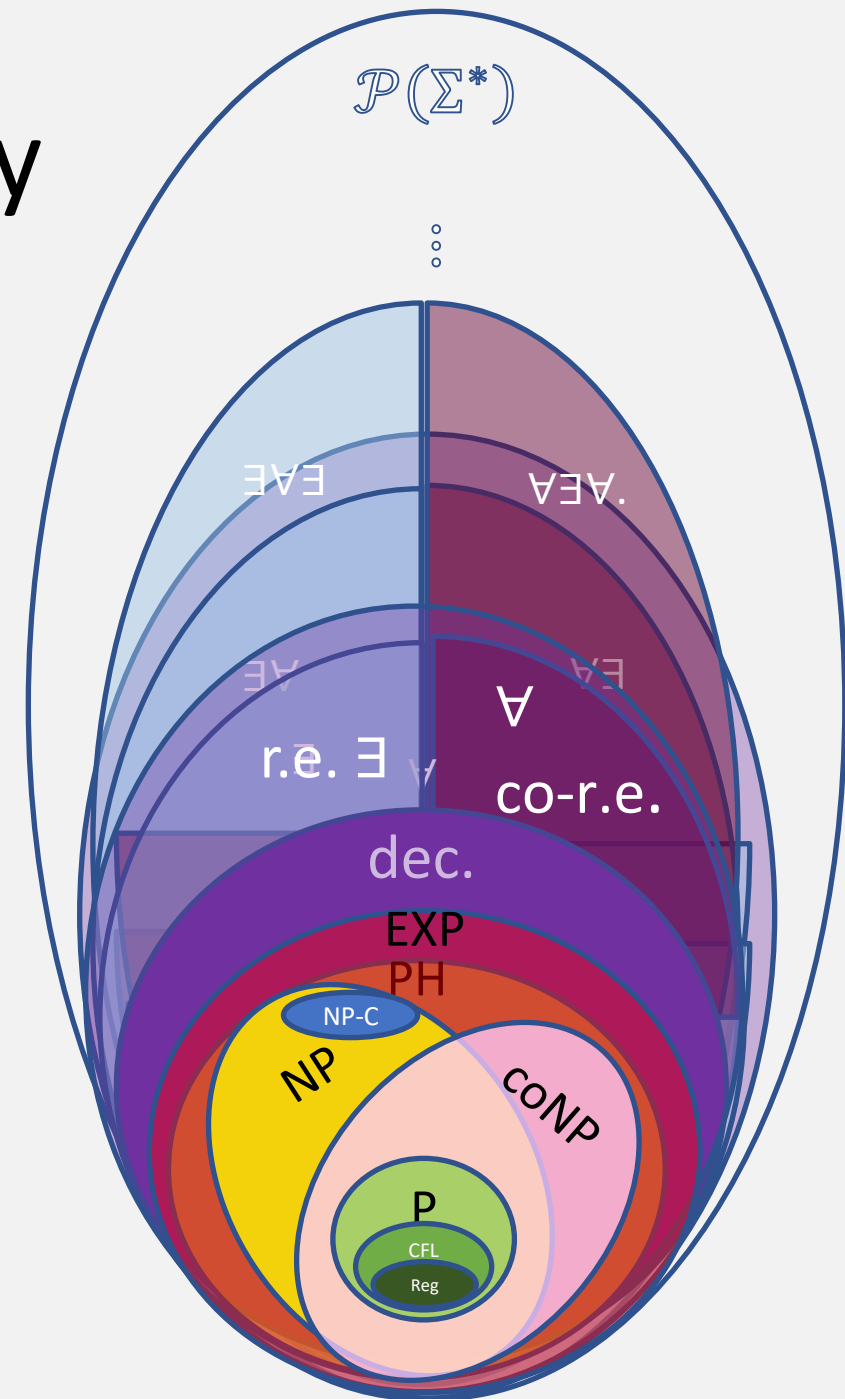


- To prove **that L is in a class, give an algorithm**
 - DFA, NFA, regular expression, CFG, PDA, TM, TM that always halts
- To prove **that L is not in a class, or as hard as anything in the class: use reductions**
 - Mapping reductions for computability
 - $A_{TM} \leq_m \overline{E_{TM}}$, $A_{TM} \leq_m INF_{TM}$,
 - Polytime reductions for complexity above P
 - $SAT \leq_p 3SAT \leq_p IndSet \leq_p CLIQUE \dots$
- To prove hardness for regular and context-free languages, use **pumping lemmas**
 - $\{0^n 1^n\}$, $\{w \mid w \text{ is a palindrome}\}$, etc are not regular (but are CFLs)
 - $\{a^n b^n c^n\}$, $\{ww\}$, etc are not CFLs
- To prove **completeness**: show L is both **in the class** and **is hard for the class**.



Topics in complexity

- Scaling down: “polynomial-time” (polytime) on all computations, polynomial bound on all quantifiers
- Two definitions of NP:
 - Quantifier-verifier: $L = \{x | \exists y, |y| \leq \text{poly}(|x|) R(x, y)\}$ where R is polytime computable
 - Non-deterministic polytime TMs
- NP is decidable in exponential time ($NP \subseteq EXP$)
 - Known: $P \neq EXP$. Known: $P \subseteq NP \subseteq PH \subseteq EXP$
 - Open: is $P = NP$? Million dollar problem!
 - Also open: $NP = coNP$? $NP = PH$? $NP = EXP$? $EXP = PH$?...
- Definition of completeness
 - A_{TM} is complete for r.e.
 - SAT is NP-complete by Cook-Levin theorem
 - Lots of languages are NP-complete
 - And lots of computational problems are NP-hard



In P

Graph completeness, Shortest path, 2SAT, Circuit Evaluation, Linear equation solvability, Primality, DFA/NFA emptiness, regular/CFL membership

In PH, not known to be in NP

- $\forall \exists 3\text{SAT}$ (in 2^{nd} level of PH):
 - $\{\langle \varphi \rangle \mid \varphi(x, y) \text{ is 3-CNF st } \forall x \exists y \varphi(x, y) \text{ is true}\}$
- MaxClique (in 2^{nd} level of PH):
 - $\{\langle G, k \rangle \mid \text{the largest complete subgraph of } G \text{ has } k \text{ vertices}\}$
- VC dimension (in 3^{rd} level of PH)
 - $\{\langle C, k \rangle \mid C \text{ encodes a list of sets st } VC(C) \geq k\}$
(For a list of sets $\mathcal{S}$, $VC(\mathcal{S})$ is the size of the largest set X such that $X \cap S$ contains all subsets of X)

NP-complete (in NP, not known to be in P)

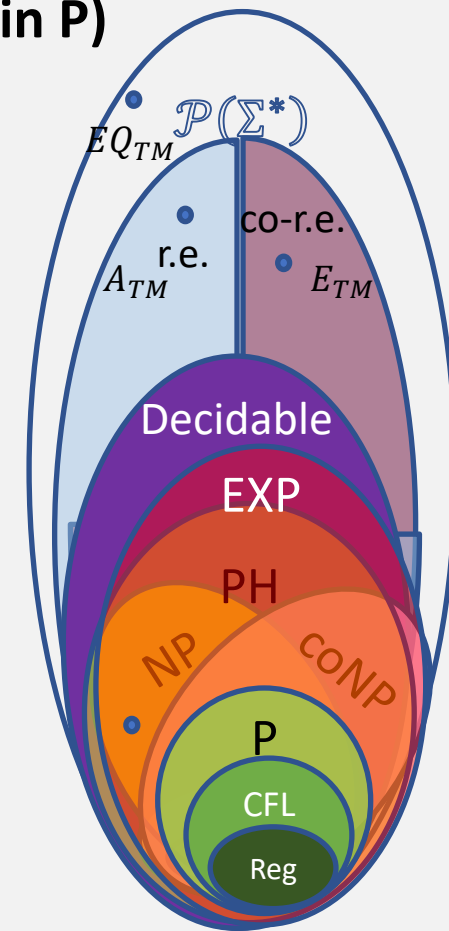
Clique, IndSet, LongestPath, SAT, 3SAT, CircuitSAT, SubsetSum, Partition, Knapsack

In EXP, not known to be in PH

- $n \times n$ checkers: 
 - $\{\langle B \rangle \mid B \text{ is a checkers board where black can guarantee a win}\}$
- Const-size TM A_{TM}^t
 - $\{\langle M, t \rangle \mid M \text{ is a TM with } |\langle M \rangle| < 100 \text{ which halts in } t \text{ steps}\}$

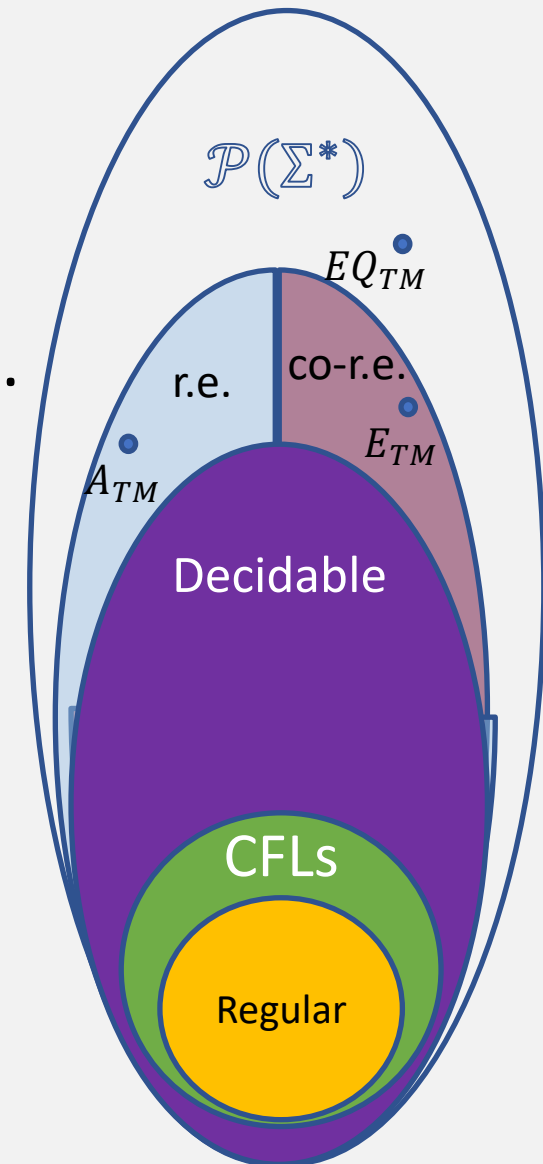
Decidable not in EXP

- Presburger arithmetic provability
 - $\{\langle \varphi \rangle \mid \varphi \text{ is provable in Presburger arithmetic}\}$



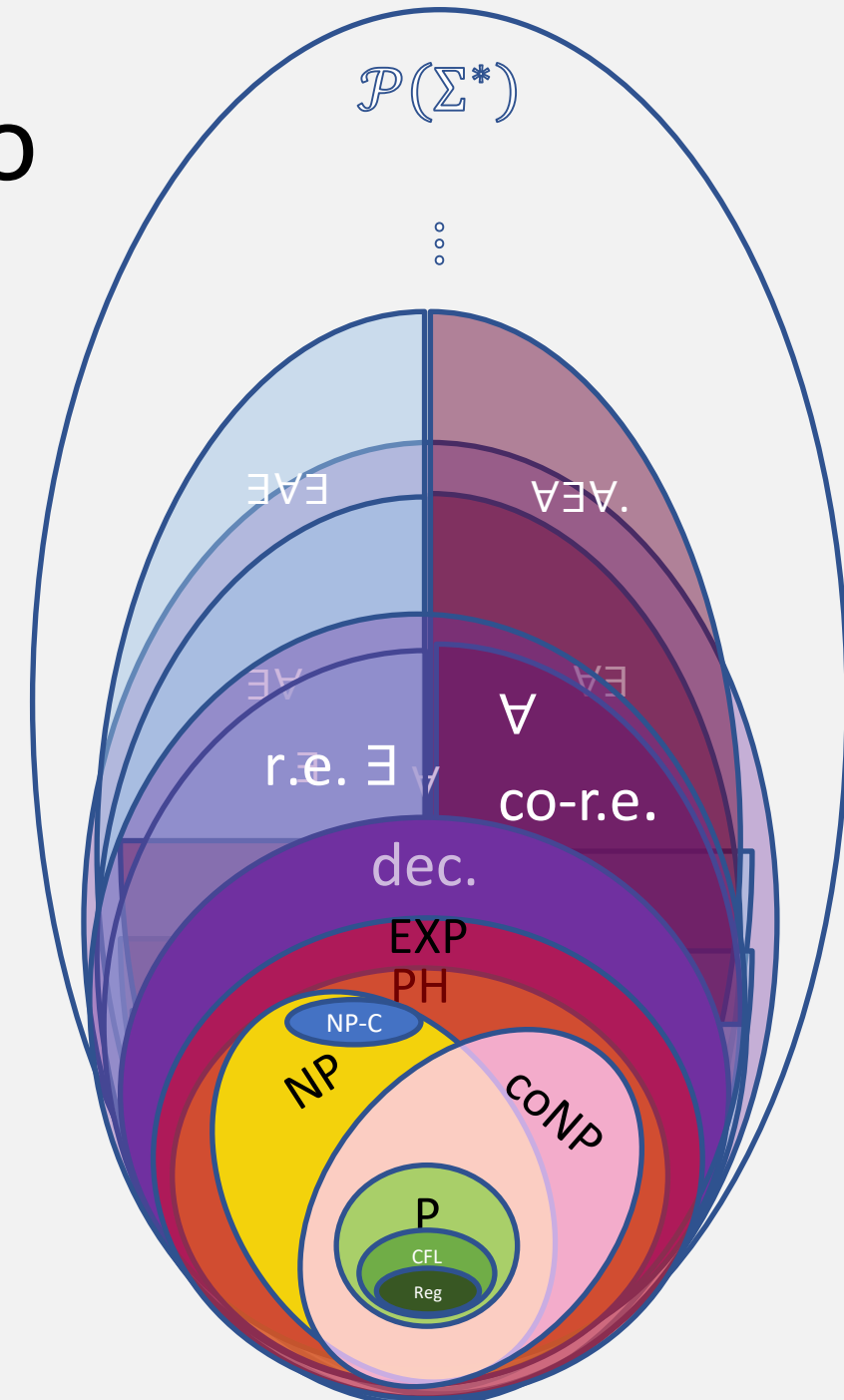
Topics in computability

- Countable and uncountable sets, diagonalization
 - Countably many TMs, but uncountably many languages.
- Undecidability, r.e, co-r.e, arithmetic hierarchy
 - $A_{TM} = \{\langle M, w \rangle \mid \exists y \text{ Check}_A(M, w, y)\}$
 - $\text{R.e.} \cap \text{co-r.e.} = \text{decidable}$.
- Proving hardness with reductions $L' \leq_m L$
 - To prove easiness, give M_L
- Recursion theorem
 - Self-print, “get own code”



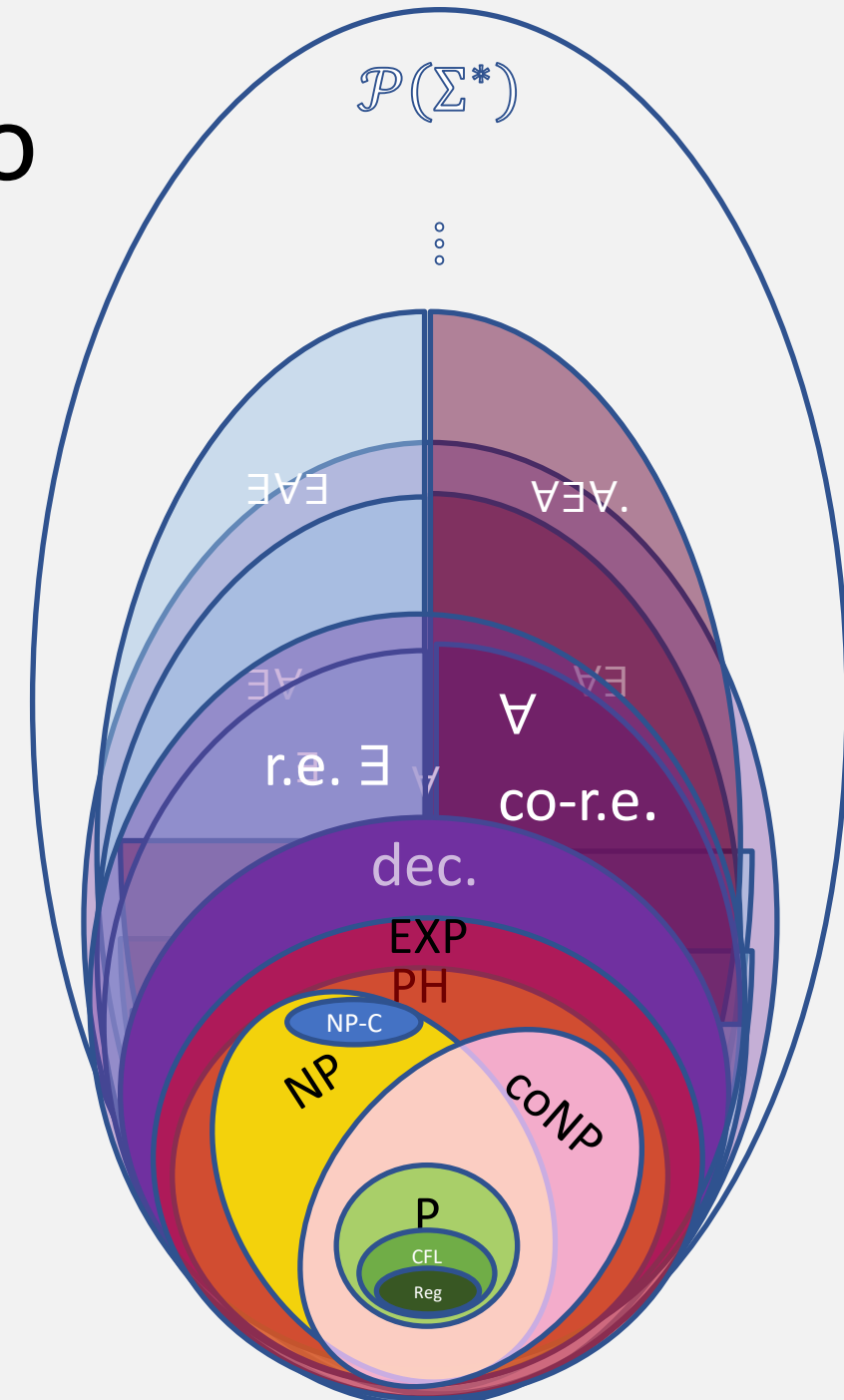
Some topics we did not get to

- Oracles:
 - An “oracle TM” has an ability to solve a specific problem (“oracle”) in unit time.
 - Another way of defining hierarchies
 - Both in computability and complexity.
 - Gives rise to another type of reductions
 - Turing reductions: $L \leq_t L'$ if L can be solved with an access to an oracle for L.
 - And its polytime analogue.



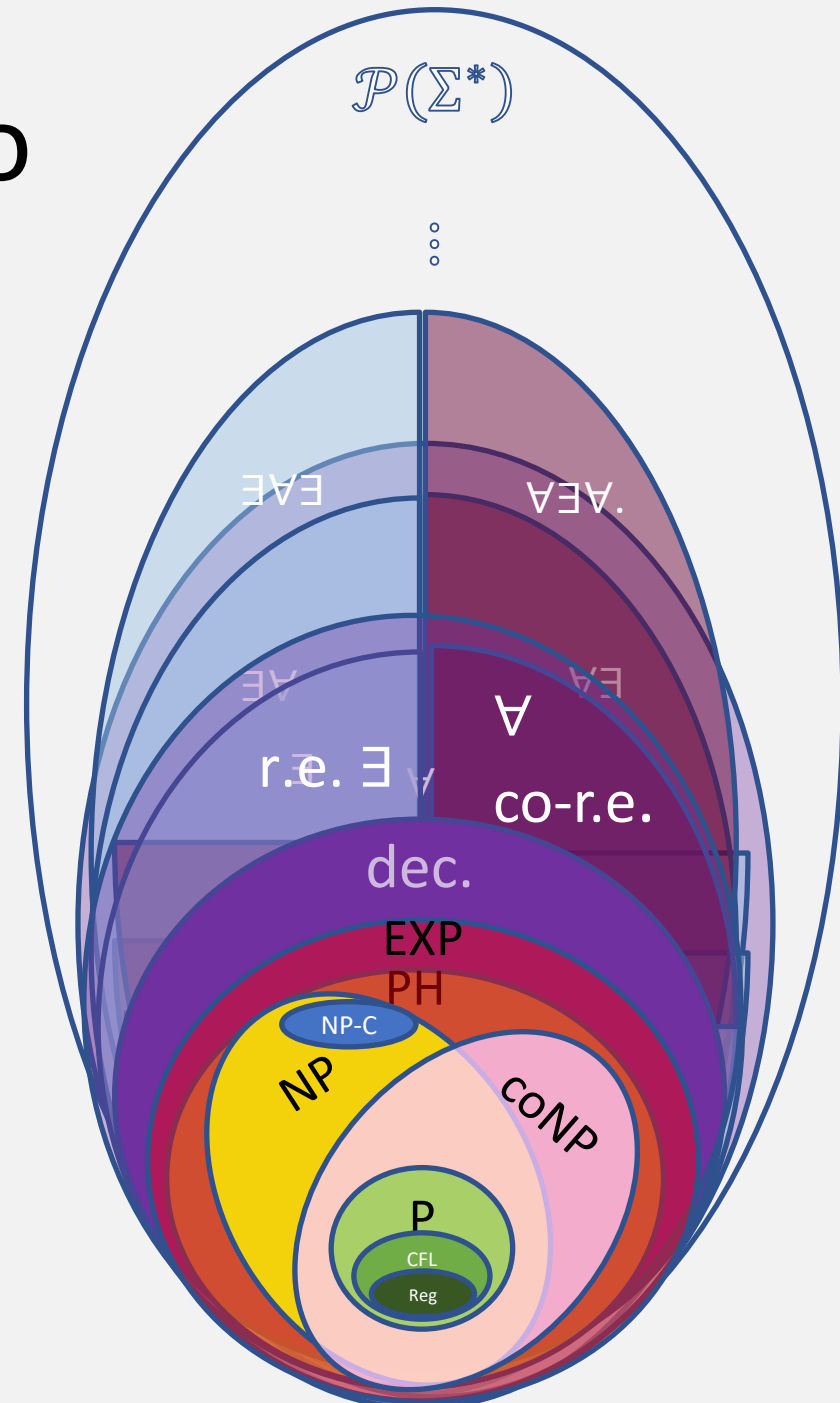
Some topics we did not get to

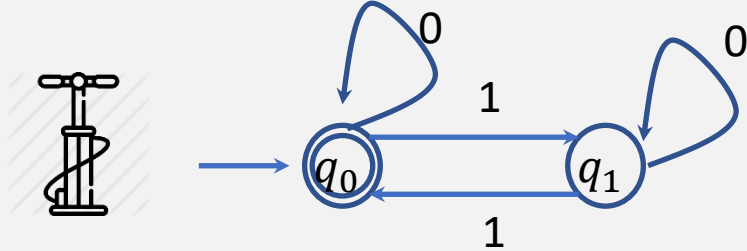
- Time complexity in models other than deterministic and non-deterministic
 - Randomized computation
 - Quantum computation
- Space-limited computation
 - Space-bounded complexity classes
 - Streaming: a model where data flows through a machine with very little memory.
 - Eg a router: can it detect things like DDoS attack?
- Computation in hardware:
 - Circuit complexity: non-uniform computation



Some topics we did not get to

- Computation as interaction
 - NP as a game
 - Probabilistically Checkable Proofs
 - Closely related to hardness of approximation
 - Interactive proofs
 - Communication complexity
 - Distributed computation
- Connections with cryptography
 - Secure primitives from hardness assumptions
- ... And many more topics, connections and open problems!





Thank you
so much!

