CMSC 28100

Introduction to Complexity Theory

Spring 2025
Instructor: William Hoza

Which problems

can be solved

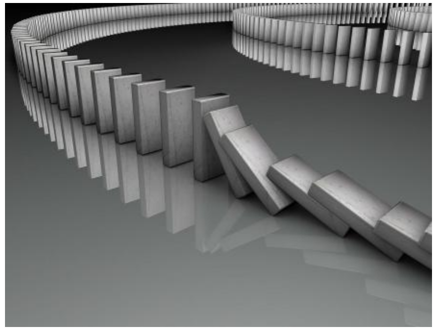
through computation?

CLASSICAL

Which languages are in $\mathbf{P}$?

Which languages are *not* in $\mathbf{P}$?

The bounded halting problem



- $\text{BOUNDED-HALT} = \{\langle M, w, T\rangle : M \text{ halts on } w \text{ within } T \text{ steps}\}$
- $\text{BOUNDED-HALT} \in \text{EXP}$

Theorem: $\text{BOUNDED-HALT} \notin \text{P}$

- Proof strategy: We'll show that if BOUNDED-HALT were in P, then it would follow that $\text{P} = \text{EXP}$

Proof that BOUNDED-HALT $\notin$ P

- Assume B is a poly-time TM deciding BOUNDED-HALT
- Let $Y \in \text{EXP}$. There is a TM M that $\begin{cases} \text{accepts } w \text{ within } 2^{|w|^k} \text{ steps} & \text{if } w \in Y \\ \text{loops} & \text{if } w \notin Y \end{cases}$
- We will construct a poly-time TM R that decides Y

R:

Given $w \in \{0,1\}^*$:

1. Simulate B on $\left\langle M, w, 2^{|w|^k} \right\rangle$
2. If B accepts, accept. If B rejects, reject.

- Polynomial time
- If $w \in Y$, then M accepts w within $2^{|w|^k}$ steps, so R accepts w
- If $w \notin Y$, then M loops on w, so R rejects w

Beyond “it’s not in P”

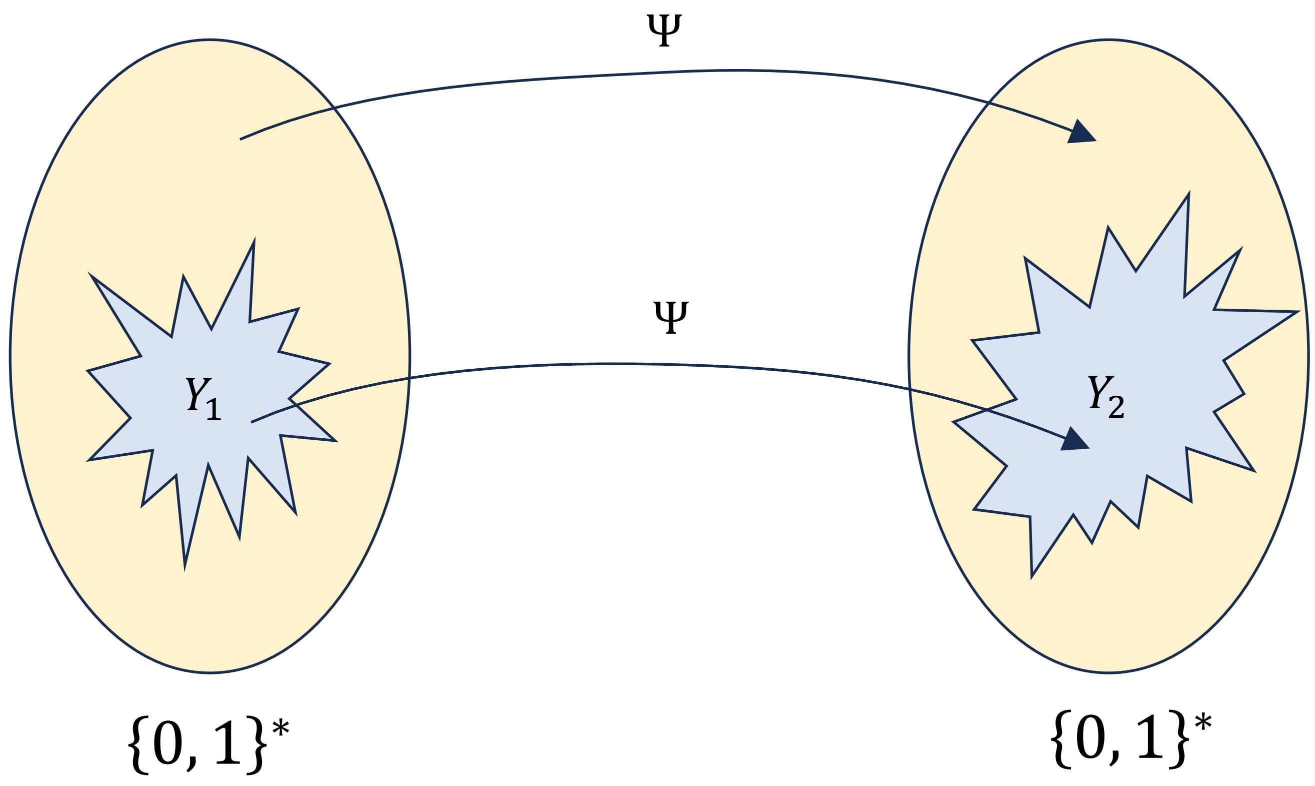
- We proved $\text{BOUNDED-HALT} \notin \mathrm{P}$
- Insight: The proof gives us bonus information
  - “How far outside P is it?”
  - “Why is it outside P? What kind of hardness does it have?”
- The proof shows that every language in EXP reduces to BOUNDED-HALT
- Furthermore, the reduction has a very specific structure

Mapping reductions

- Let $Y_1, Y_2 \subseteq \{0,1\}^*$
- **Definition:** We say that Y_1 is poly-time mapping reducible to Y_2 if there exists a poly-time TM Ψ such that for every $w \in \{0,1\}^*$:
  - If $w \in Y_1$, then Ψ halts on w with some $w' \in Y_2$ written on its tape
  - If $w \notin Y_1$, then Ψ halts on w with some $w' \notin Y_2$ written on its tape
- Notation: $Y_1 \leq_{\mathrm{P}} Y_2$
  - Intuition: "Complexity of Y_1" $\leq$ "Complexity of Y_2"

Mapping reductions

- $Y_1 \leq_{\mathrm{P}} Y_2$ means there is an efficient way to convert questions of the form "is $w \in Y_1$?" into questions of the form "is $w' \in Y_2$?"



Mapping reduction example

- $\text{COMPOSITES} = \{\langle K \rangle : K \text{ is a composite number}\}$
- $\text{FACTOR} = \{\langle K, M \rangle : K \text{ has a prime factor } p \leq M\}$
- **Claim:** $\text{COMPOSITES} \leq_{\text{P}} \text{FACTOR}$
- **Proof:** Given $\langle K \rangle$, the reduction produces $\langle K, K - 1 \rangle$. Poly-time ✔
- If K is composite, then K has a prime factor less than K ✔
- If K is not composite, then K does not have a prime factor less than K ✔

Let $n = |w|$ and $m = |w'|$. What is the relationship between n and m?

A: $m \le \text{poly}(n)$

B: $n \le \text{poly}(m)$

C: $n = m$

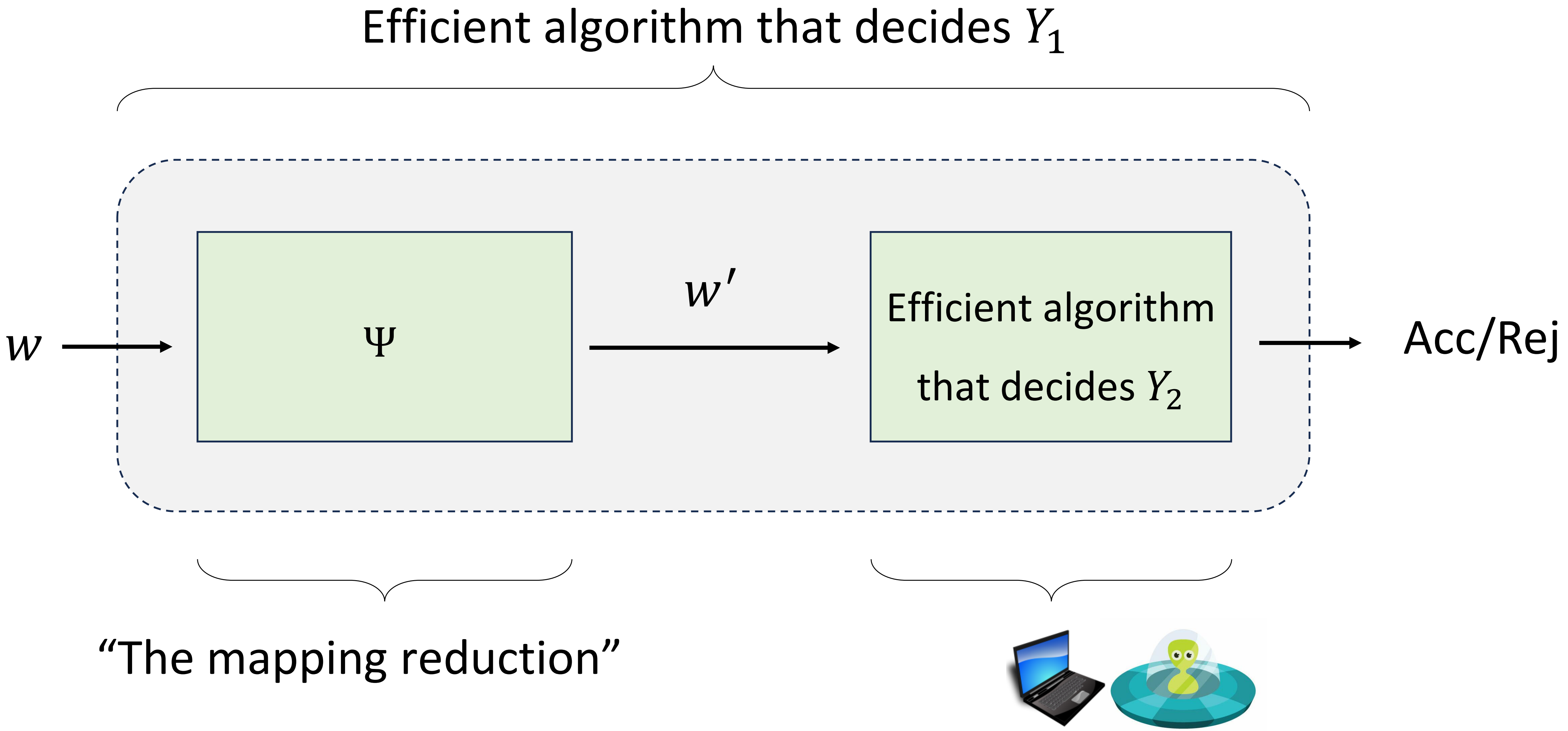
D: Not enough information

Respond at PollEv.com/whoza or text “whoza” to 22333

guage is in P

- **Proof:** Given $w \in \{0,1\}^*$:
  1. Simulate Ψ to produce w' (this takes $O(n^{k_1})$ time)
  2. Check whether $w' \in Y_2$ (this takes $O(m^{k_2})$ time where $m = |w'|$)
  3. If so, accept; otherwise, reject.
- $m \le O(n^{k_1})$, so the total time is $O(n^{k_1} + n^{k_1 \cdot k_2}) = \text{poly}(n)$

Reductions: Proving that a language is in P



Reductions: Proving that a language is not in P

- Let $Y_1, Y_2 \subseteq \{0,1\}^*$
- **Claim:** If $Y_1 \leq_{\mathrm{P}} Y_2$ and $Y_1 \notin \mathrm{P}$, then $Y_2 \notin \mathrm{P}$
- **Proof:** If Y_2 were in P, then Y_1 would also be in P

EXP-hardness

- Let $Y \subseteq \{0, 1\}^*$
- **Definition:** We say that Y is "EXP-hard" if, for every $L \in \mathrm{EXP}$, we have $L \leq_{\mathrm{P}} Y$
- Interpretation:
  - Y is at least as hard as any language in EXP
  - Every problem in EXP is basically a special case of Y

Example: $\mathrm{BOUNDED\text{-}HALT}$ is EXP-hard

- **Claim:** $\mathrm{BOUNDED\text{-}HALT}$ is EXP-hard
- **Proof:** Let $Y \in \mathrm{EXP}$. We will show $Y \leq_{\mathrm{P}} \mathrm{BOUNDED\text{-}HALT}$
- Let M be a 2^{n^k}-time TM deciding Y
- Construct M' by replacing q_{reject} with a looping state
- Mapping reduction Ψ: Given w, construct $\left\langle M', w, 2^{|w|^k} \right\rangle$

EXP-hard languages are intractable

- Let $Y \subseteq \{0,1\}^*$
- **Claim:** If Y is EXP-hard, then $Y \notin \mathrm{P}$
- **Proof:** There exists $Y_{\mathrm{hard}} \in \mathrm{EXP}$ such that $Y_{\mathrm{hard}} \notin \mathrm{P}$
- Since Y is EXP-hard, we have $Y_{\mathrm{hard}} \leq_{\mathrm{P}} Y$

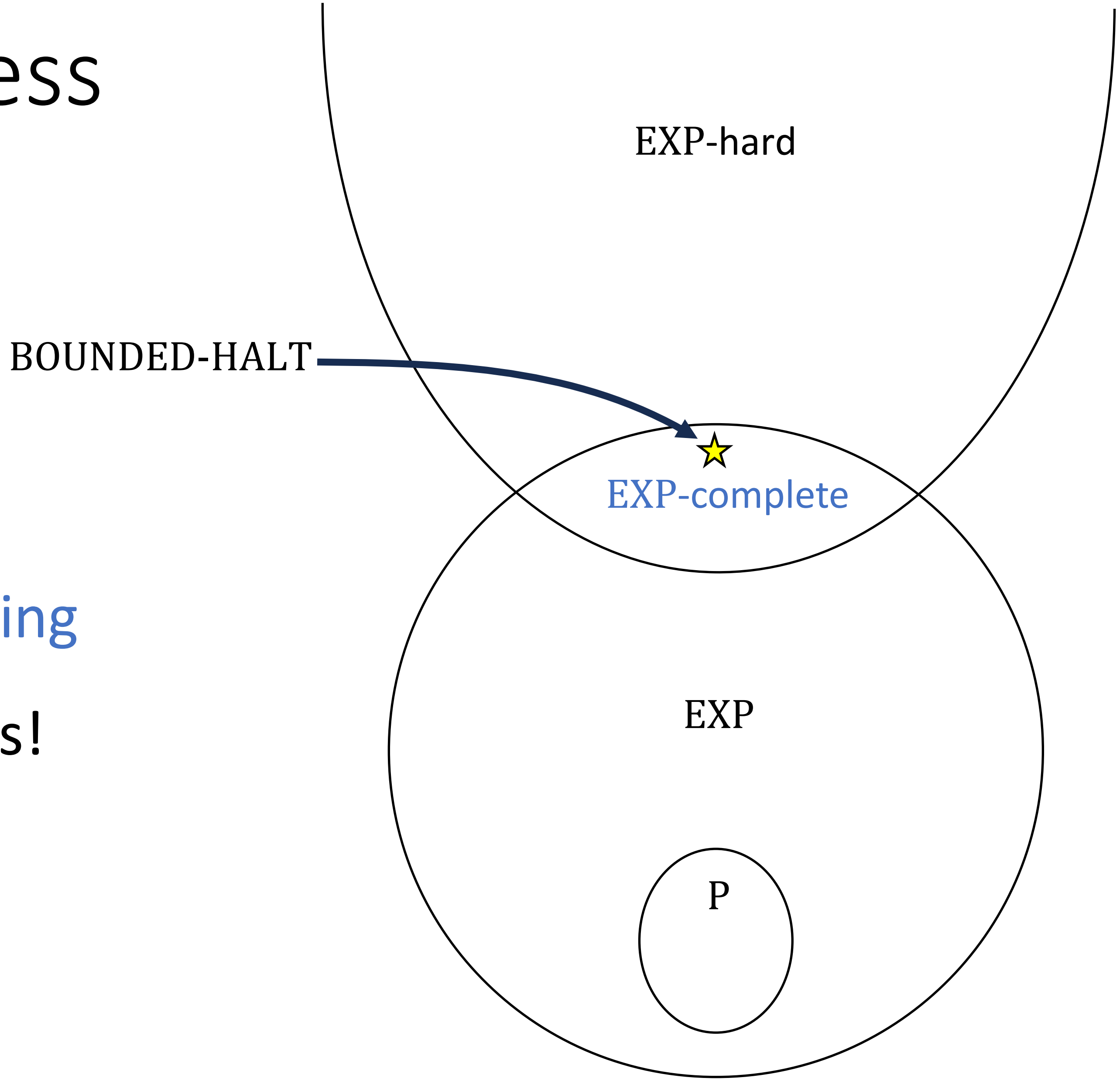
EXP-completeness

- Let $Y \subseteq \{0,1\}^*$
- **Definition:** We say that Y is EXP-complete if Y is EXP-hard and $Y \in \mathrm{EXP}$
- The EXP-complete languages are the hardest languages in EXP
- If Y is EXP-complete, then the language Y can be said to "capture" / "express" the entire complexity class EXP

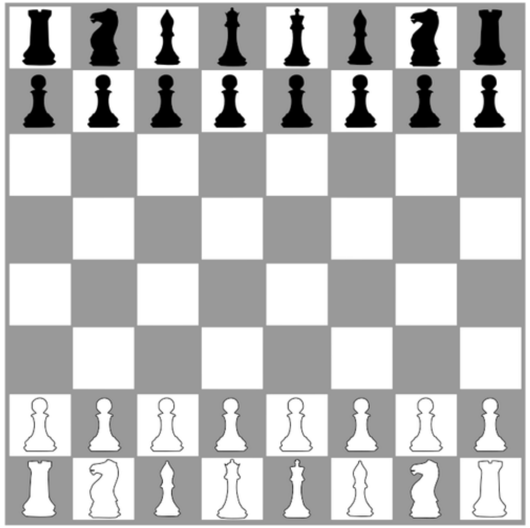
EXP-completeness

There are many interesting

EXP-complete languages!



Example: Chess



- Let GENERALIZED-CHESS $= \{\langle P \rangle : P$ is an arrangement of chess pieces on an $N \times N$ board from which "white" can force a win$\}$

Theorem: GENERALIZED-CHESS is EXP-complete.

Consequently, GENERALIZED-CHESS $\notin$ P.

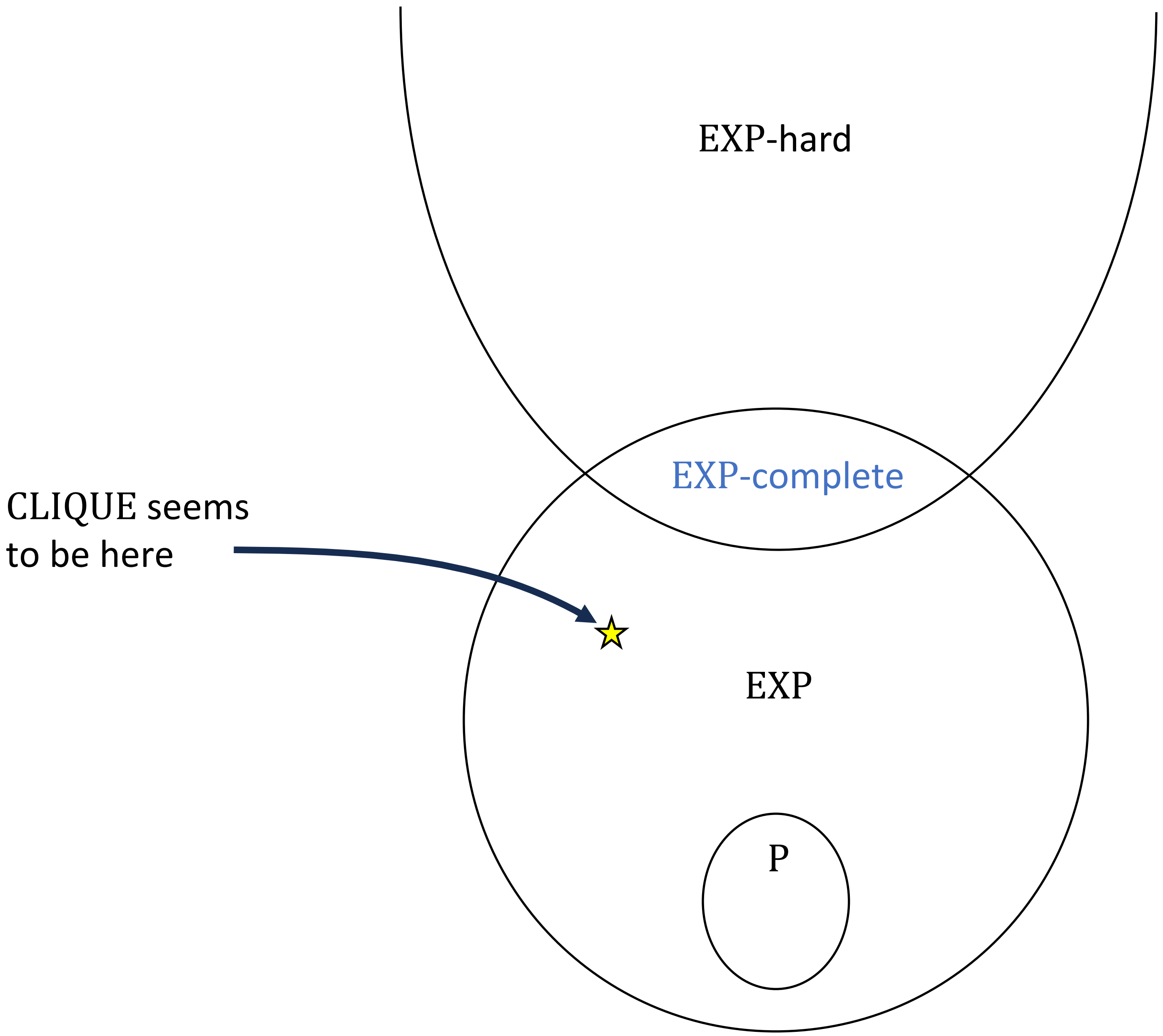
- (Proof omitted. This theorem will not be on exercises/exams)

The power of $\mathbf{EXP}$-hardness

- EXP-hardness is a valuable tool for identifying intractability
- Is EXP-hardness the only tool we need for identifying intractability?

Complexity of the clique problem

- Recall $\mathrm{CLIQUE} = \{\langle G, k \rangle : G \text{ has a } k\text{-clique}\}$
- $\mathrm{CLIQUE} \in \mathrm{EXP}$. (Why?)
- If you spend a while trying to design a good algorithm, eventually you might start to suspect that $\mathrm{CLIQUE} \notin \mathrm{P}$
- However, if you spend a while trying to design a good reduction, eventually you might start to suspect that CLIQUE is not EXP-complete either!


EXP-hard
EXP-complete
CLIQUE seems
to be here
EXP
P

Complexity of the clique problem

- Evidently, to understand the complexity of CLIQUE, we need new conceptual tools

Guessing and checking

- **Key insight:** There exists a polynomial-time randomized Turing machine M with the following properties.
  - If $\langle G, k\rangle \notin \mathrm{CLIQUE}$, then $\Pr[M \text{ accepts } \langle G, k\rangle] = 0$.
  - If $\langle G, k\rangle \in \mathrm{CLIQUE}$, then $\Pr[M \text{ accepts } \langle G, k\rangle] \neq 0$.

"Nondeterministic TM"

- **Proof:** M picks a random subset of the vertices, accepts if it is a k-clique, and rejects otherwise.