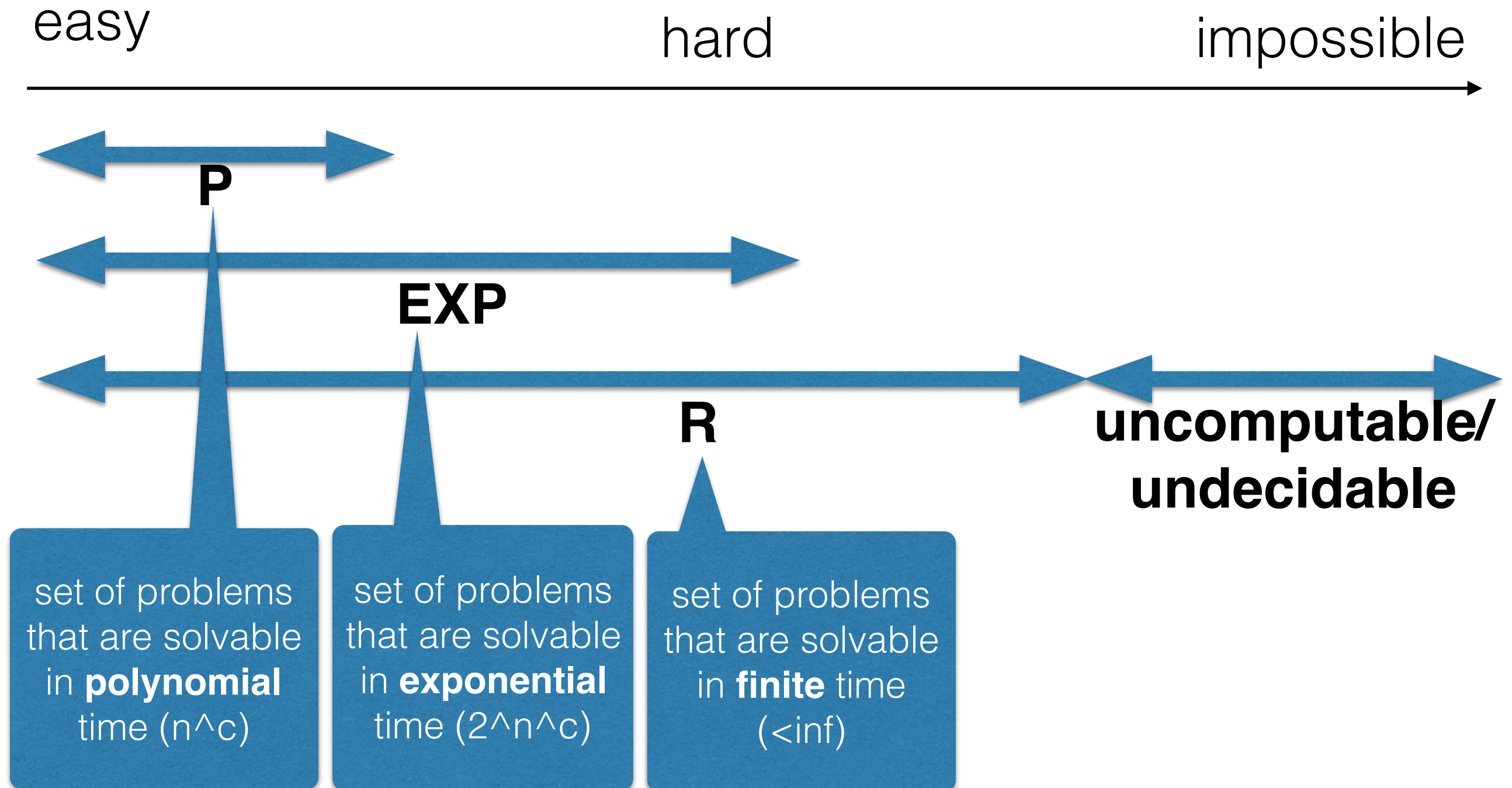


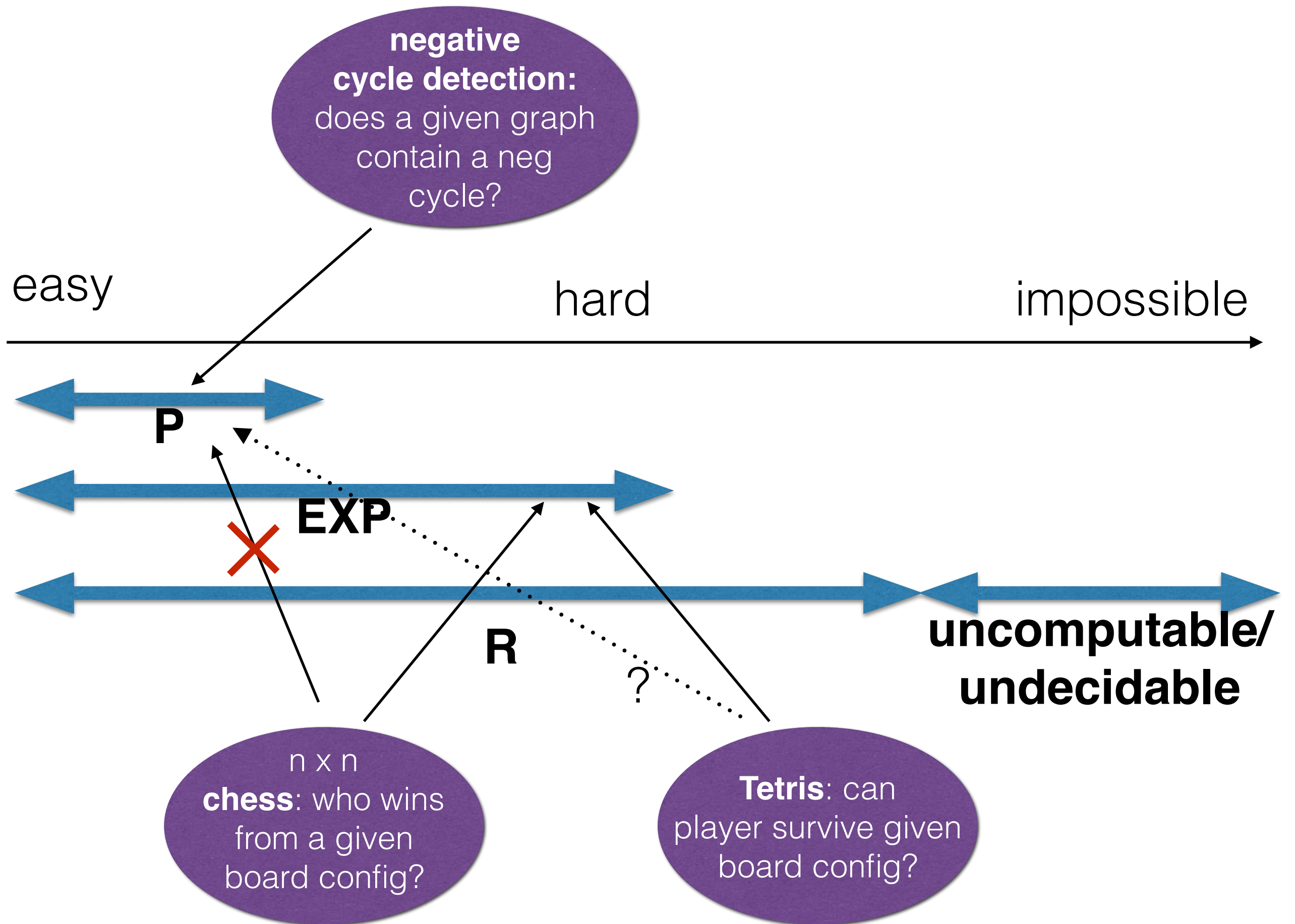
Computational complexity

CS 146 - Spring 2017

Which is harder, Tetris or Chess?

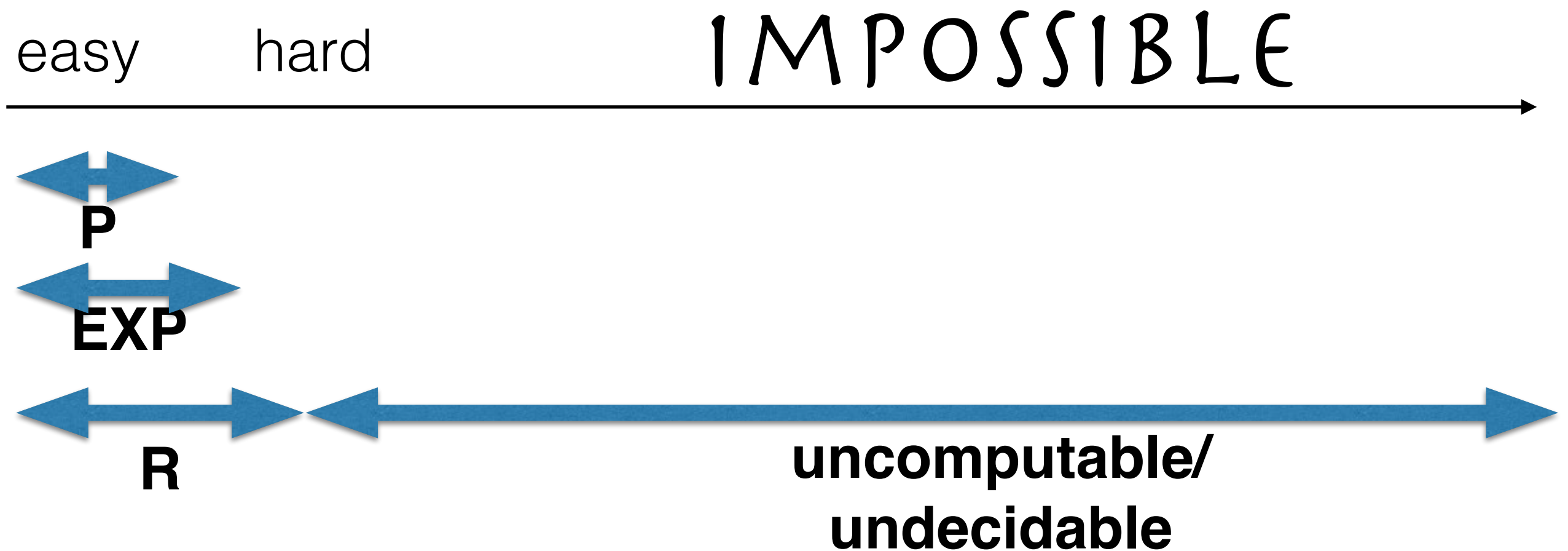
Computational difficulty





[Breukelaar, Demaine, Hohenberger, Hoogeboom, Koster, Liben-Nowell 2004]

Most decision problems are uncomputable



Proof sketch: ...

Decision problem:
YES/NO question

VS

Optimization problem:
find the min/max question

negative cycle detection

maximum subarray sum

is the minimum change $< x$ coins?

find the minimum change

is the max knapsack value $> \$x$?

knapsack

single-source
shortest paths

finding the MST

knapsack
(hw 12)

easy

hard

impossible

P

EXP

R

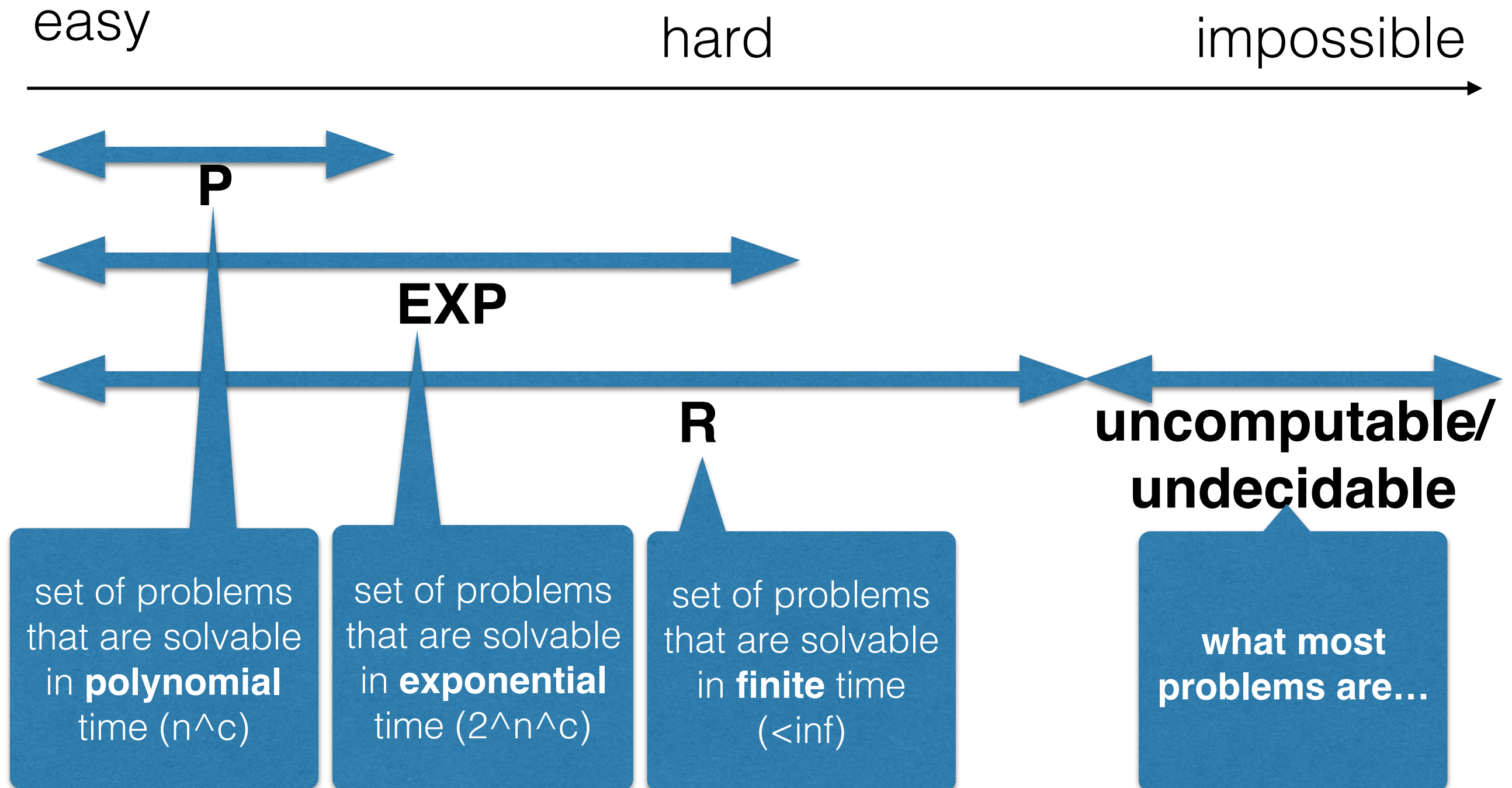
**uncomputable/
undecidable**

Make change

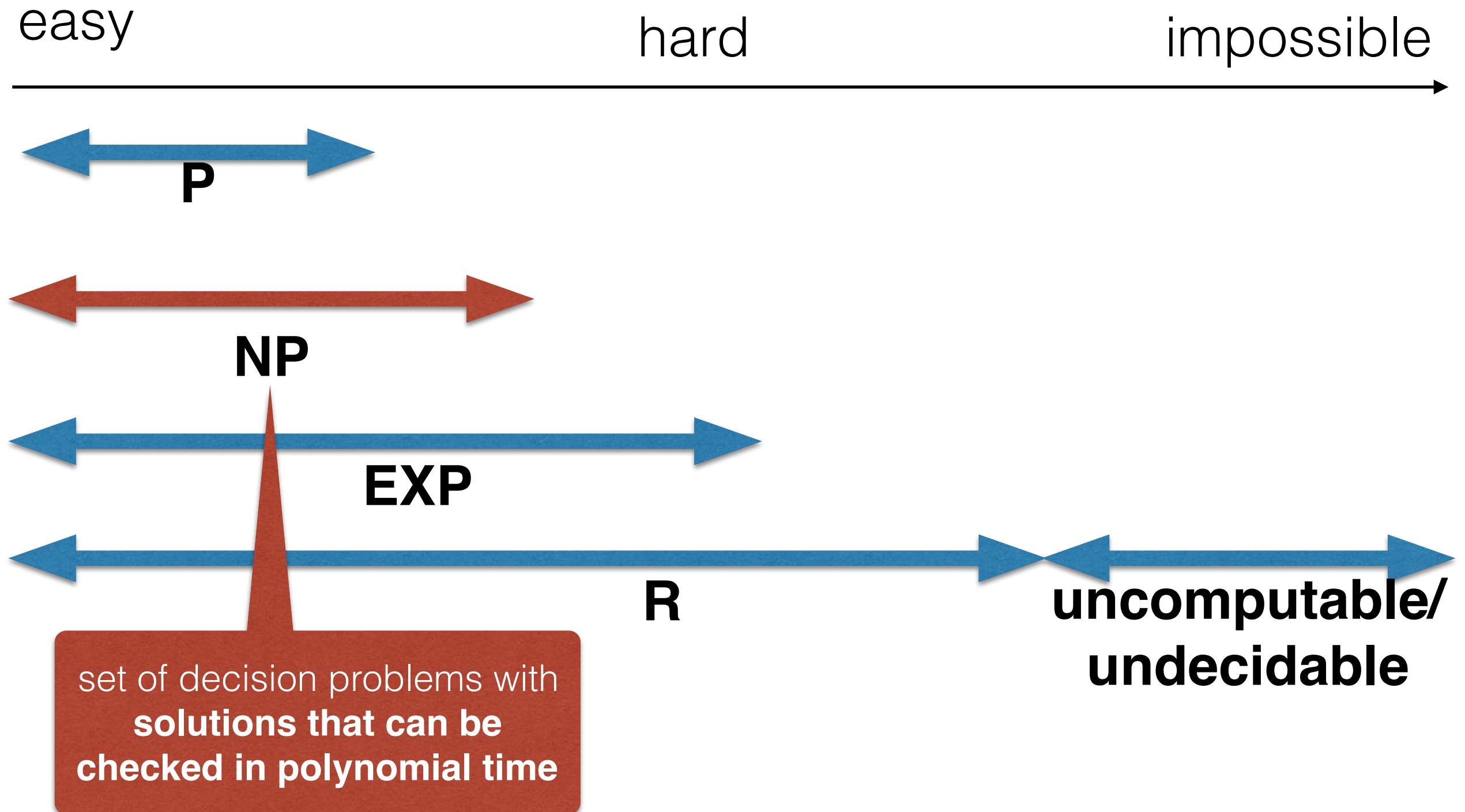
longest
increasing
subsequence

longest common
subsequence

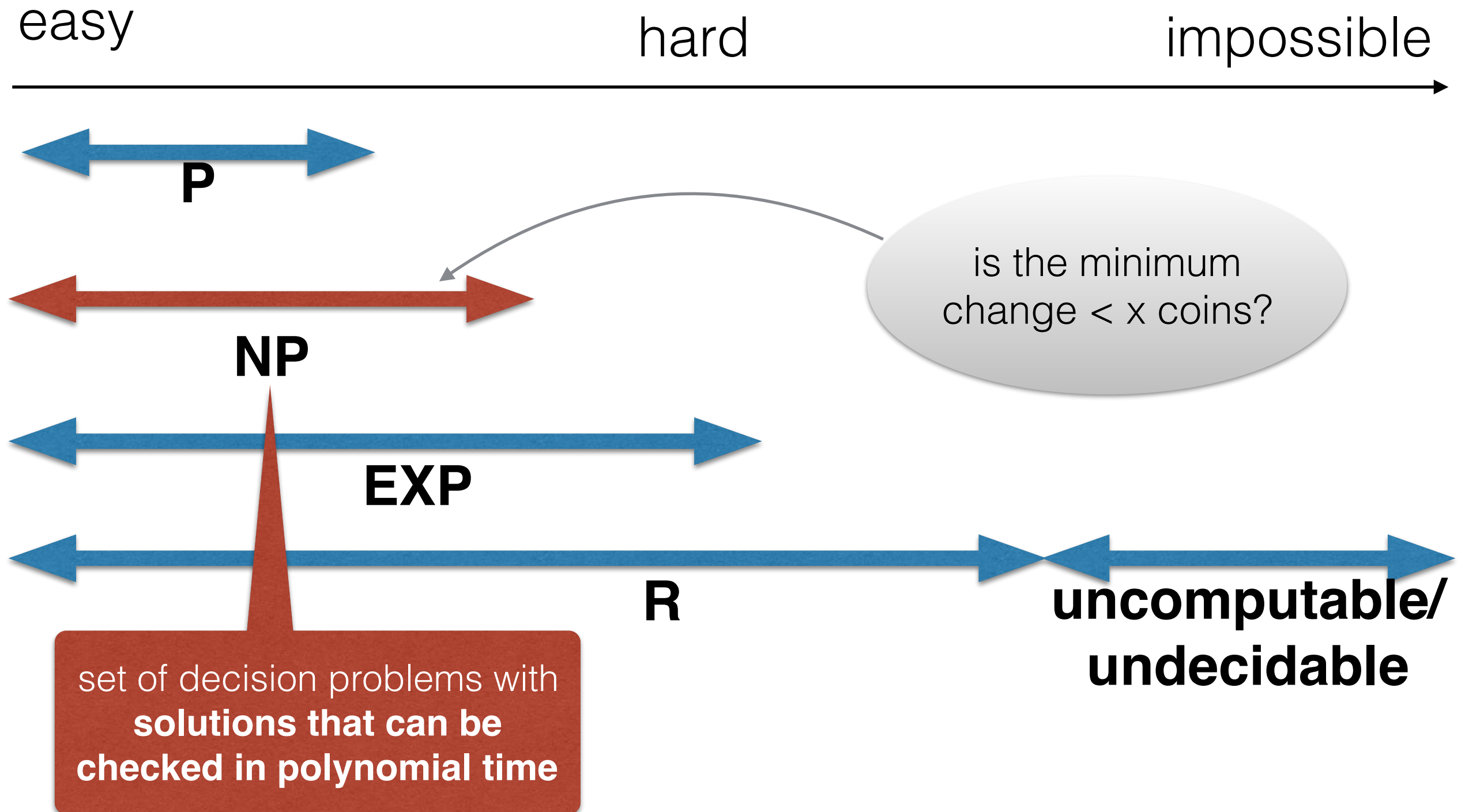
Computational difficulty



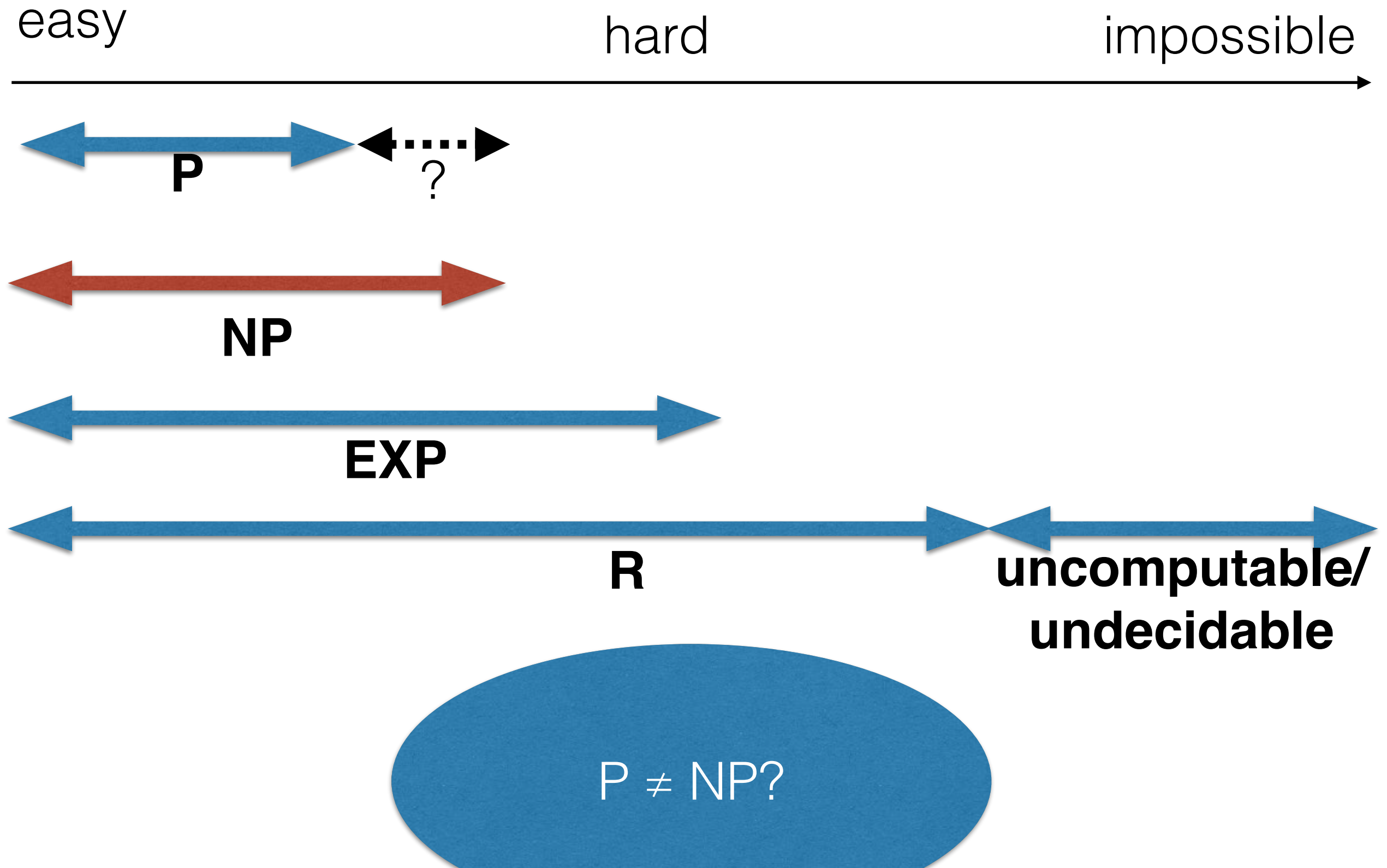
Computational difficulty



Computational difficulty



\$1,000,000 conjecture

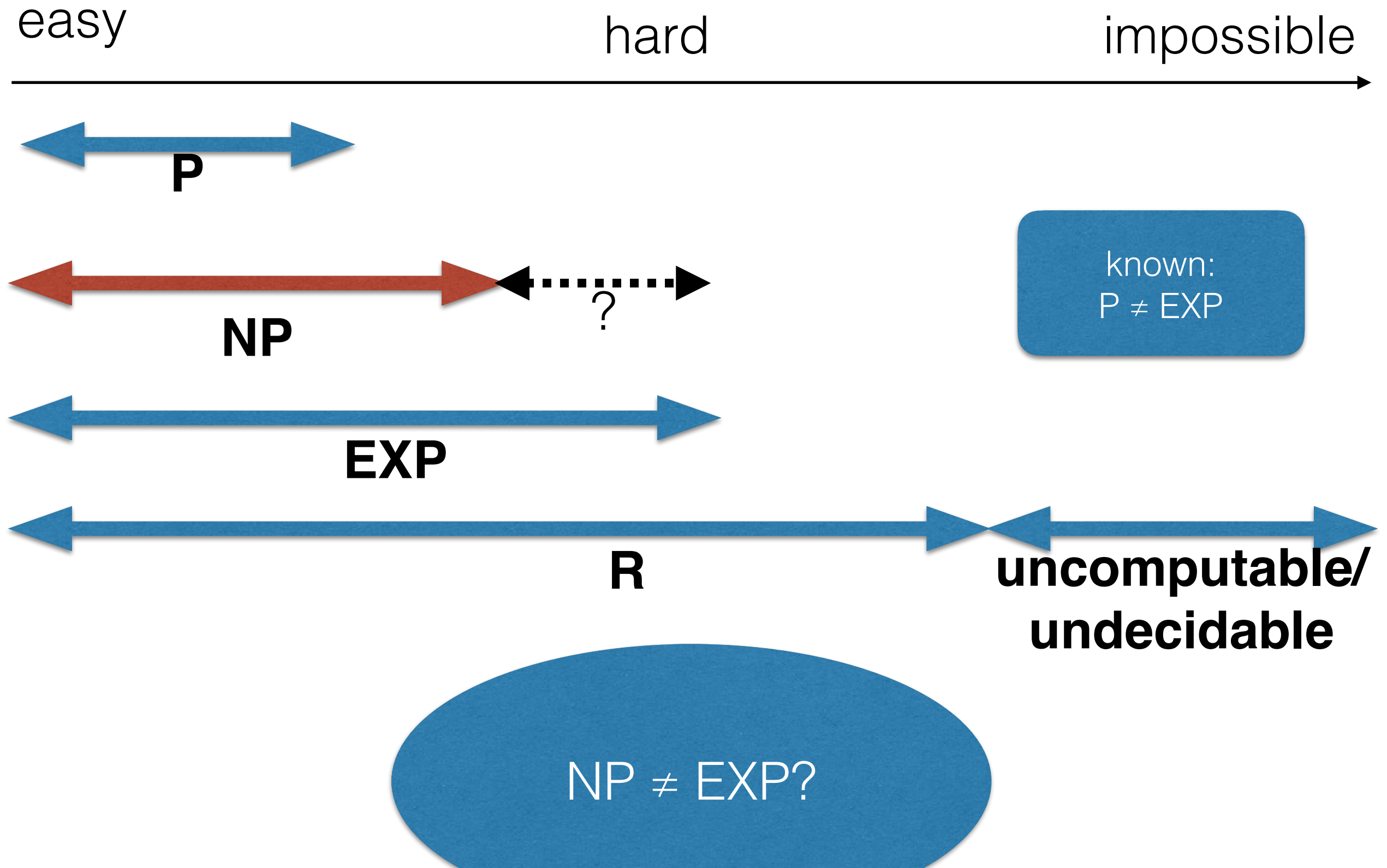


Warning!

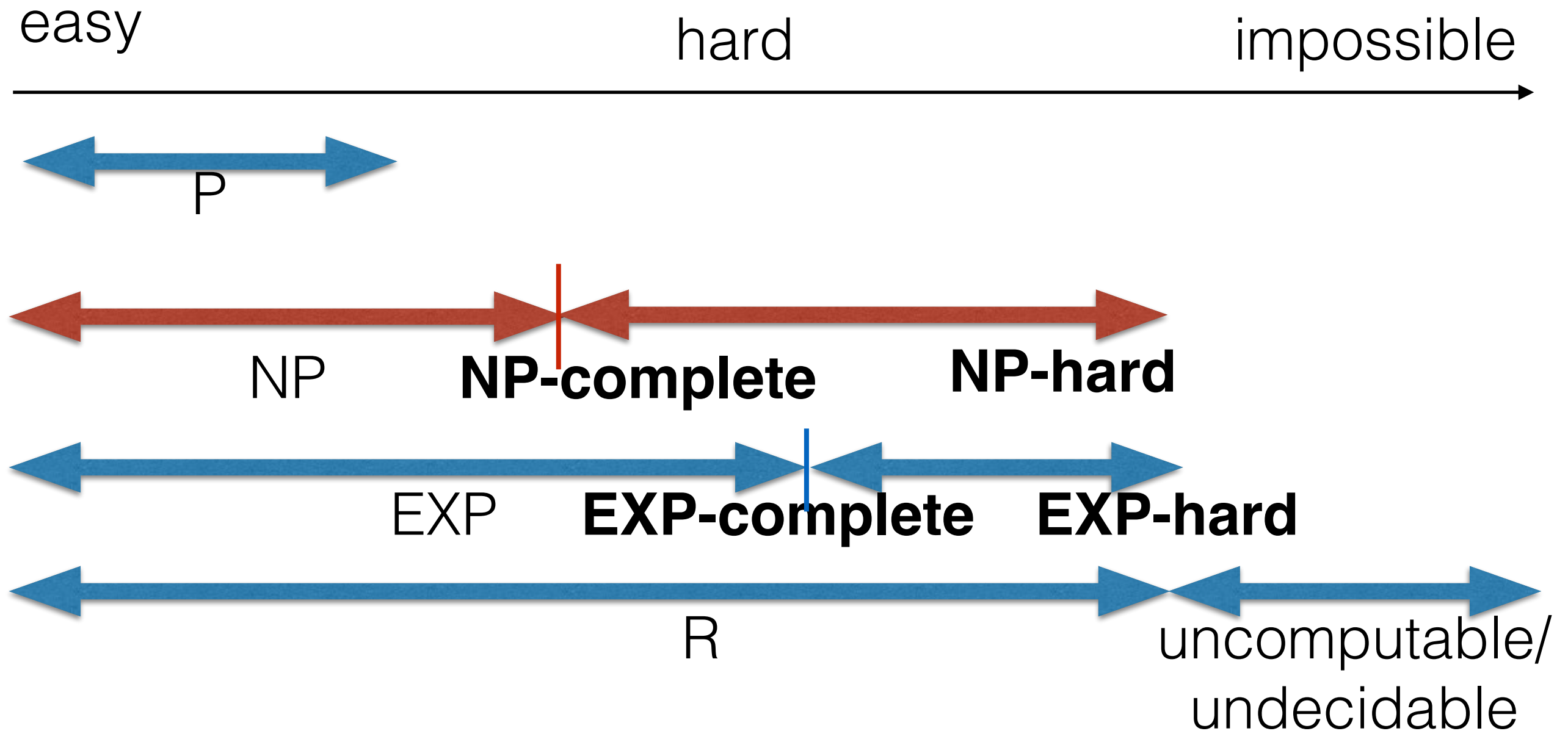
NP stands for **N**on-deterministic
Polynomial

(not non-polynomial)

Also open, but no money



Completeness, hardness

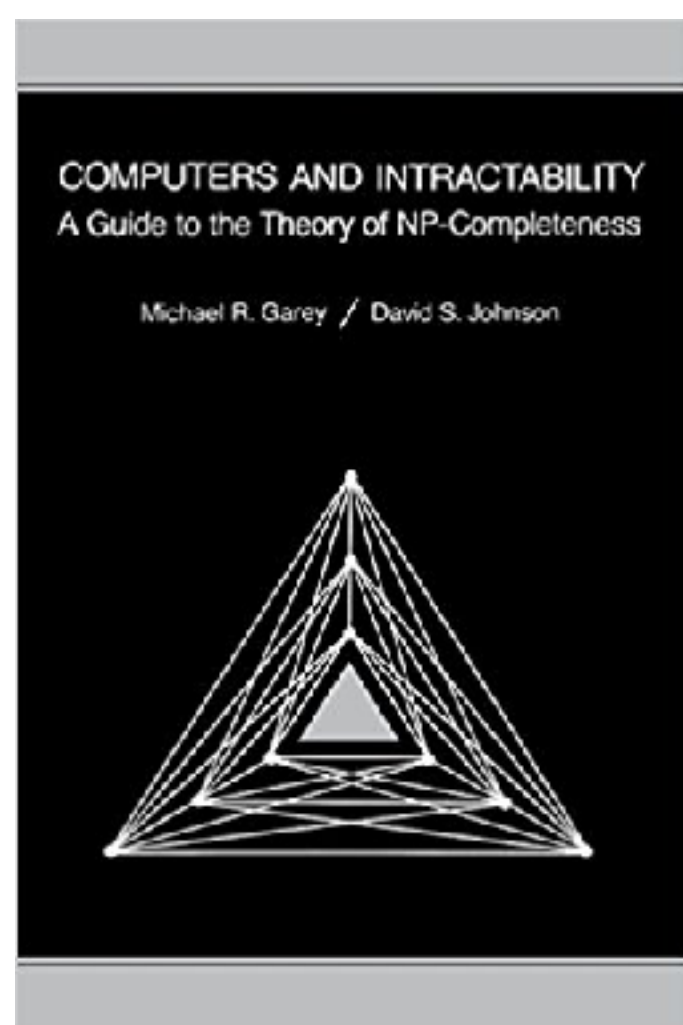
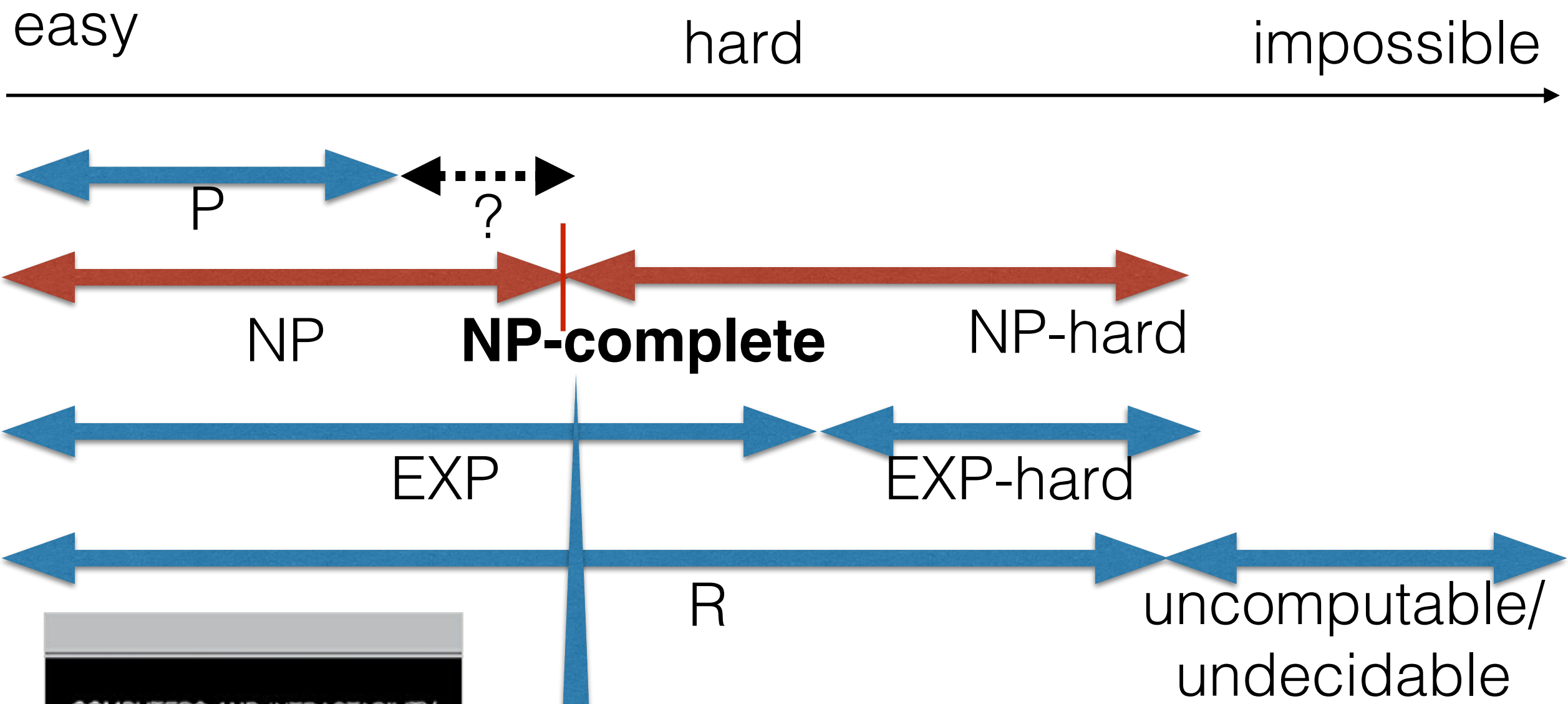


EXP-hard: as hard as any problem in EXP

EXP-complete: in EXP and as hard as any problem in EXP

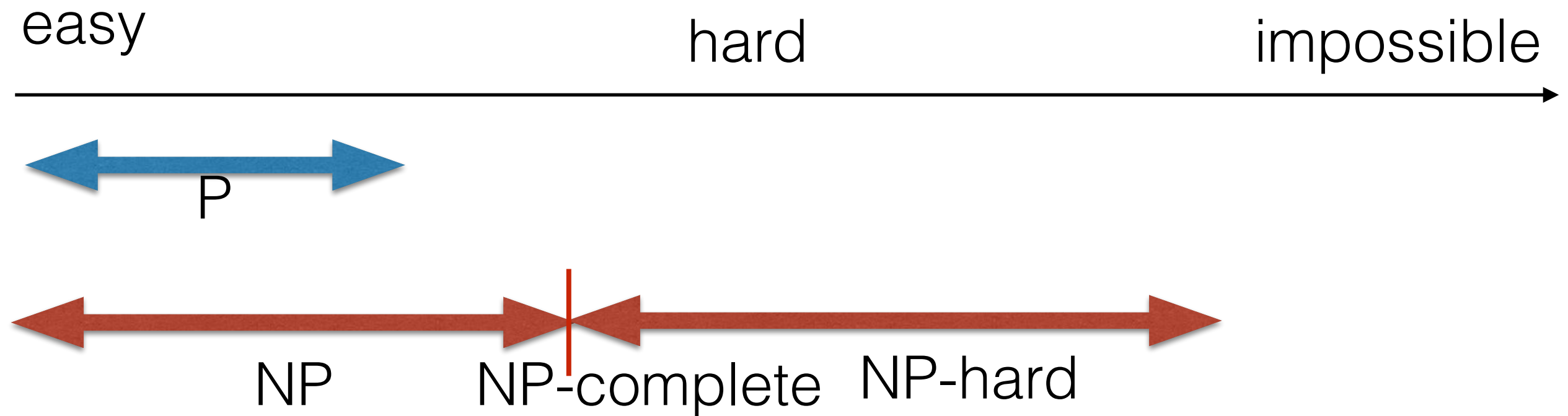
Reduction

- convert your problem into a problem you already know how to solve, instead of solving your problem from scratch



a lot of problems were found to be NP-complete

More open problems



almost all interesting problems
are known to be in P
or are NP-complete

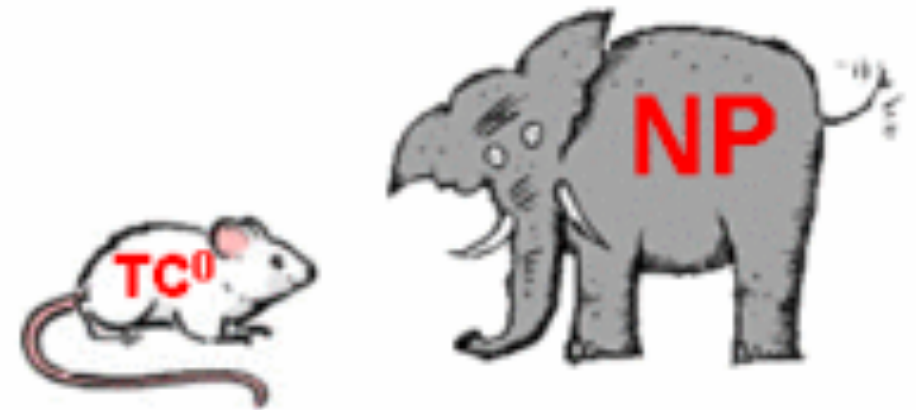
except...

is factoring in P?

is graph isomorphism in P?

But wait, there's more...

<http://complexityzoo.com>



→ exponential-time algorithms

what to do then? → approximation algorithms

→ fixed-parameter tractable algorithms

→ heuristic search

Acknowledgements

lecture based on Erik Demaine's

<http://courses.csail.mit.edu/6.006/fall11/notes.shtml>