11. Tractability and Decidability

Frank Stephan

April 1, 2014

Computational Complexity

In 1844 Gabriel Lamé studied and analysed the running time of Euclid's algorithm. In 1982, Stephen Cook received the Turing Award for his contribution to complexity of computation. In 1985 Richard Karp received the Turing Award for his contributions to the theory of algorithms.

Complexity Theory

"The foundations [for complexity theory] were laid in the 1930s by a group of logicians, including Alan Turing, who were concerned with the existence or nonexistence of automatic procedures for deciding whether mathematical statements were true or false." Karp's Turing Award Lecture, *Combinatorics, Complexity, and Randomness* (http://www.jdl.ac.cn/turing/pdf/p98-karp.pdf)

Decidability

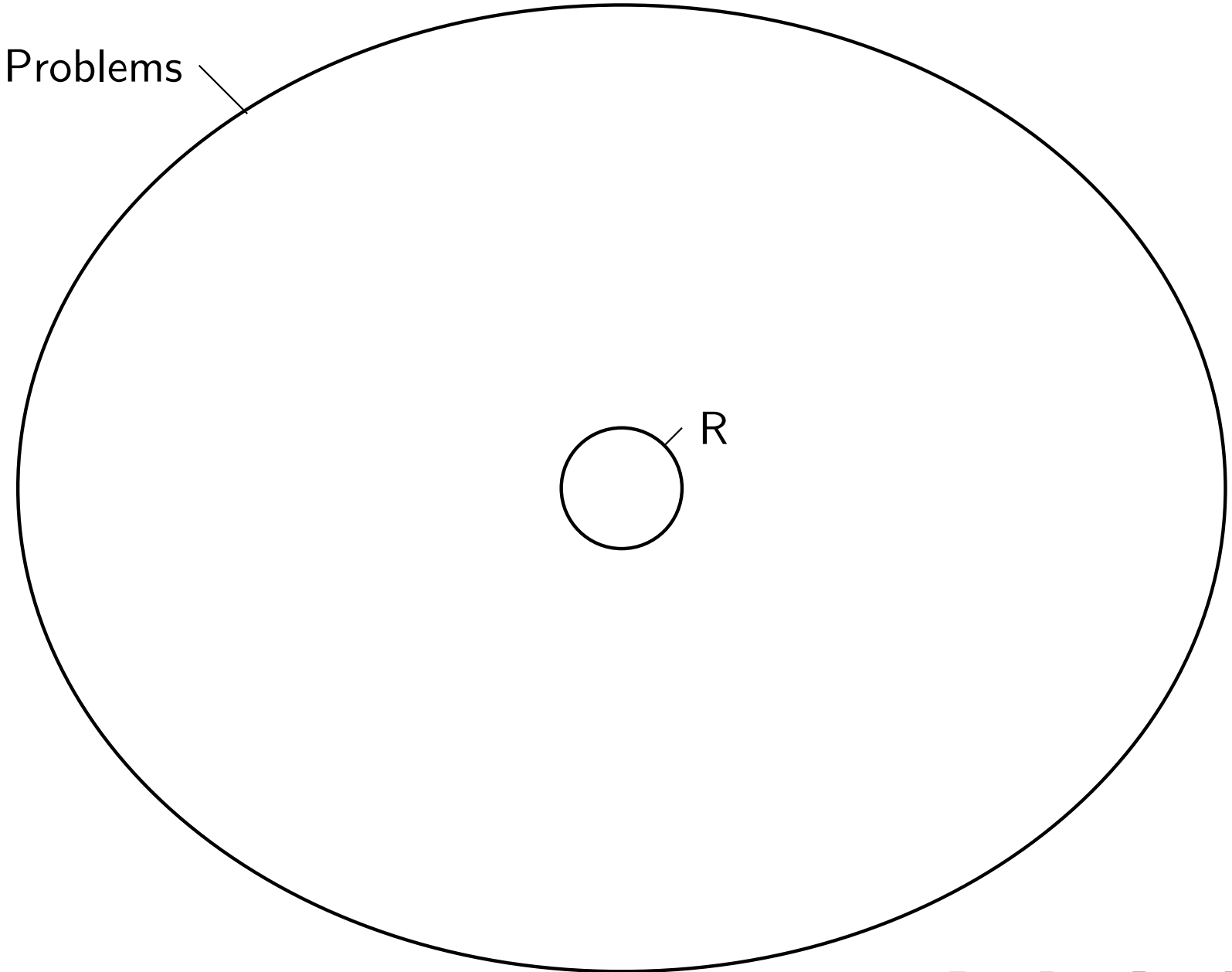
A problem is decidable if there exists an algorithm that can solve all instances of the problem. We call R the class of all decidable problems.

Undecidability

A problem is undecidable if there does not exist an algorithm that can solve all instances of the problem.

There are many more Problems than Algorithms

Most problems are undecidable.


Problems
R

Cardinality

- The set $\emptyset$ has 0 elements.
- The set $\{Cantor\}$ and the set $\{Turing\}$ have 1 element.
- The set $\{Cantor, Hilbert, Gödel\}$ and the set $\{Turing, VonNeumann, Karp\}$ have 3 elements.

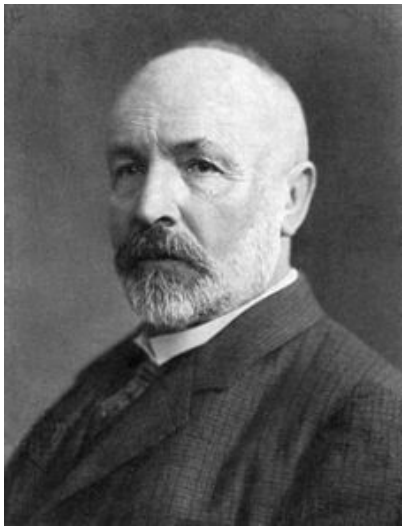
Natural Numbers

$\mathbb{N} = \{0, 1, 2, 3, 4, \cdots\}$ is the set of possible cardinalities of finite sets.

Infinity

There are infinitely many natural numbers. The number of natural numbers is a not a natural number. There are $\aleph_0$ (pronounce aleph-naught, aleph-null, or aleph-zero) (countably many, countably infinite) natural numbers (they are countable).

$$\aleph_0$$



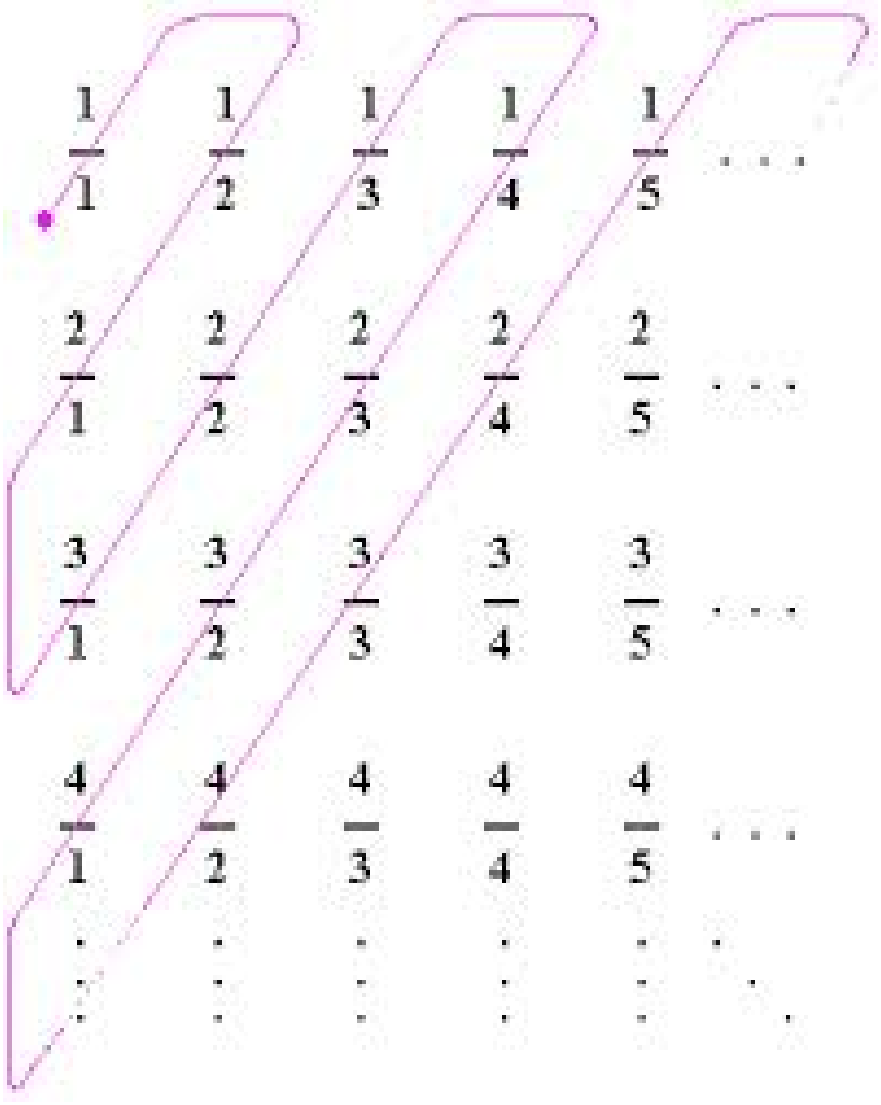
Countability

The set of odd natural numbers is $\{1, 3, 5, 7, \cdots\}$. There are as many odd numbers as there are natural numbers. There are $\aleph_0$ (countably many) natural numbers.

$$m_{odd} = 2 \times n + 1$$

Countability

The set of rational numbers is $\{0, \frac{1}{3}, \frac{1}{2}, 1, \frac{5}{4}, \cdots\}$. There are as many rational numbers as there are natural numbers. There are $\aleph_0$ (countably many) natural numbers.


1/1 1/2 1/3 1/4 1/5 . . .
2/1 2/2 2/3 2/4 2/5 . . .
3/1 3/2 3/3 3/4 3/5 . . .
4/1 4/2 4/3 4/4 4/5 . . .
. . . . . .
. . . . . .
. . . . . .

Sets of Sets

The set $\{1, 2, 3\}$ contains $2^3 = 8$ subsets of numbers: $\emptyset$, $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$, $\{1, 2, 3\}$

Sets of Natural Numbers

How many sets of natural numbers are there?

$$2^{\aleph_0}$$

$$2^{\aleph_0} = \aleph_1? \quad \text{(Continuum Hypothesis)}$$

Sets of Natural Numbers

Let us represent a set of natural number as a (infinite) vector.

$$\emptyset =< 0, 0, 0, 0, 0, 0, 0, \ldots >$$

$$\{0, 1\} =< 1, 1, 0, 0, 0, 0, 0, \ldots >$$

$$\{2, 4, 5\} =< 0, 0, 1, 0, 1, 1, 0, \ldots >$$

set /number	0	1	2	3	4	⋯
$\emptyset$	0	0	0	0	0	⋯
$\{0\}$	1	0	0	0	0	⋯
$\{1,2\}$	0	1	1	0	0	⋯
$\{4\}$	0	0	0	0	1	⋯
⋯	⋯	⋯	⋯	⋯	⋯	⋯
$\{0,1\}$	1	1	0	0	0	⋯
⋯	⋯	⋯	⋯	⋯	⋯	⋯

set /number	0	1	2	3	4	$\cdots$
$\emptyset$	$\underline{0}$	0	0	0	0	$\cdots$
$\{0\}$	1	$\underline{0}$	0	0	0	$\cdots$
$\{1,2\}$	0	1	$\underline{1}$	0	0	$\cdots$
$\{4\}$	0	0	0	$\underline{0}$	1	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\underline{\cdots}$	$\cdots$
$\{0,1\}$	1	1	0	0	0	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\underline{\cdots}$

Diagonal Argument

Let us select a set A which differs from the diagonal everywhere:

$$< 0, 0, 1, 0, \cdots, \cdots > \rightarrow A = < 1, 1, 0, 1, \cdots, \cdots >$$

The new set A is distinct from all rows.

Proof

If there was a row identical to A, say the n^{th} row, then the n^{th} entry of A would be identical to the n^{th} entry in the n^{th} row, although A differs at the n^{th} entry from the n^{th} entry of the diagonal. Thus A does not occur in the list.

Algorithms are Countable

An algorithm can be represented by pseudo-code, by a program in JavaScript, by the program input to a universal Turing machine, namely, by a finite string of 1s and 0s.

An algorithm is a natural number.

Problems are Uncountable

A decision problem is a function from a set of inputs to the set $\{yes, no\}$. There are $\aleph_0$ possible inputs and $2^{\aleph_0}$ many possible answer-vectors to these inputs. A decision problem is the subset of those possible inputs which have yes as an answer. There are $2^{\aleph_0}$ problems.

There are $2^{\aleph_0}$ problems, $2^{\aleph_0}$ subsets of $\mathbb{N}$ and $2^{\aleph_0}$ real numbers.

Example: Primality

The primality problem seeks to decide whether a natural number is a prime number.

An algorithm is a natural number: it can be represented by the finite sequence of bits of its JavaScript code.

$$< 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, \cdots, 1, 0 >$$

A problem is a real number: for each natural number, the following infinite vector indicates whether it is a prime or not.

$$< 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, \cdots >$$

Do we Know Undecidable Problems?

The halting problem, Post correspondence problem, Wang tiles, Diophantine equations, etc.

The Halting Problem is Undecidable

There is no algorithm that can decide whether an algorithm halts on every input.

Proof (by Turing)

Assume that there exists a function `Halt(F, i)` that decides whether the function `F` halts on input `i`. It returns 1 if `F` halts on input `i`, and 0 otherwise.
Let us consider the function `X` below.
If we call `X(X)`, the call will loop forever if and only if Halt says that `X` with input `X` halts. This is a contradiction.

```
1 function X(i){
2     if (Halt(i,i) == 0) then return 0;
3     else loop forever;}
```

Post Correspondence Problem

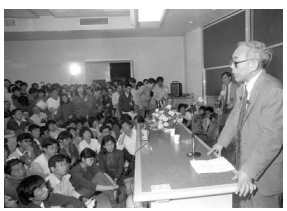
Given a finite list of pairs of words, is there a sequence of any length of these pairs possibly repeated that yields the same sentence on the first and second line?

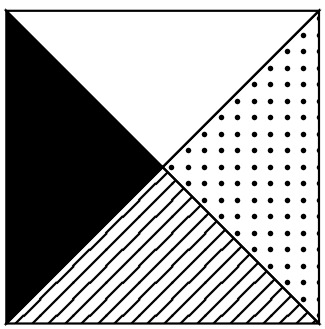
$$\begin{pmatrix} \text{s} \\ \text{Amer} \end{pmatrix} \begin{pmatrix} \text{a} \\ \text{ic} \end{pmatrix} \begin{pmatrix} \text{il} \\ \text{l} \end{pmatrix} \begin{pmatrix} \text{Po} \\ \text{Pos} \end{pmatrix} \begin{pmatrix} \text{wa} \\ \text{an} \end{pmatrix} \begin{pmatrix} \text{s} \\ \text{t} \end{pmatrix}$$

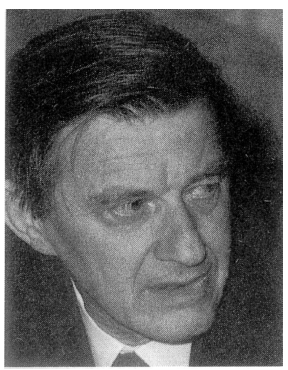
$$\begin{pmatrix} \text{Em} \\ \text{Emi} \end{pmatrix} \begin{pmatrix} \text{t} \\ \text{was} \end{pmatrix} \begin{pmatrix} \text{n} \\ \text{an} \end{pmatrix} \begin{pmatrix} \text{A} \\ \text{l} \end{pmatrix} \begin{pmatrix} \text{merican logici} \\ \text{og} \end{pmatrix}$$

Wang Tiles

A Wang tile is square split into four equilateral triangles painted in colors. As proposed by Wang Hao in 1961, the Wang tiles problem is the question whether a finite set of tiles can tile the plane such that adjacent triangles from different tiles have the same color. Wang thought the problem was decidable. In 1966, Robert Berger proved the problem is undecidable.





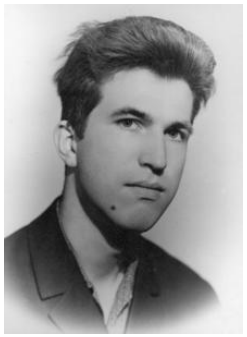


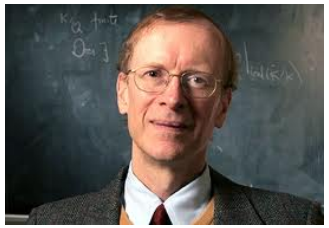
Diophantine equations

A Diophantine equation is a multivariable polynomial equation whose solutions should be integers. Deciding whether any given Diophantine equation has a solution is David Hilbert's tenth problem. In 1970, Yuri Matiyasevich proved that it is undecidable.

Fermat's Last Theorem

Andrew Wiles proved in 1995, as Pierre de Fermat suspected in 1637, that there is no positive integer solution to $x^n + y^n = z^n$ except for $n = 1$ and for $n = 2$ (Pythagorean triples).





Is everything solvable?

Time (1984): "Put the right kind of software into a computer and it will do whatever you want it to. There may be limits on what you can do with the machines themselves but there are no limits on what you can do with software."
David Harel (1993 and 2012): "Bullshit!"

Karps' Turing Award Citation

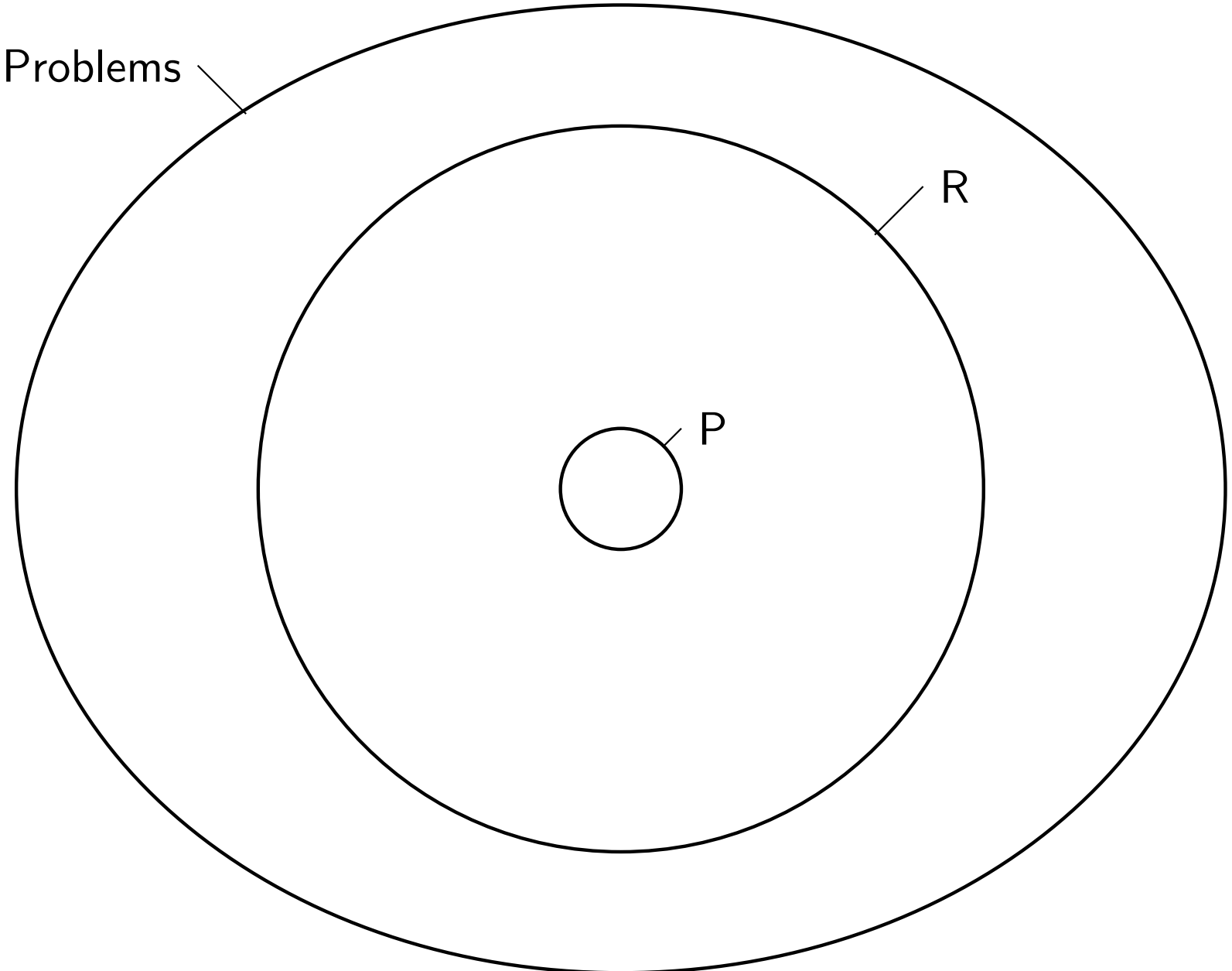
"For his continuing contributions to the theory of algorithms including the development of efficient algorithms for network flow and other combinatorial optimization problems, the identification of polynomial-time computability with the intuitive notion of algorithmic efficiency"

P

P is the set of problems that can be solved in polynomial time ($O(p(n))$ where $p(n)$ is a polynomial).

Examples

Searching a list, finding peaks, computing the maximum, sorting, Euclid algorithm, primality testing (2004 by Manindra Agrawal Neeraj Kayal and Nitin Saxena), circuit value problem, context-free grammar membership (membership in a language recognised by some push down automaton), some famous data-compression algorithms (Lempel-Ziv-Welch, 1978 and 1984).


Problems
R
P

Tractability

The Cobham thesis proposes to call the problems in P tractable and the others untractable.

Alan Cobham discussed the concept of the complexity class P in a 1965 paper.

Combinatorial Explosion

"The input to the program was a set of Boolean formulas specifying how the outputs of the circuit were to depend on the inputs: the program was supposed to generate a circuit to do the job using a minimum number of logic gates'. [...] The number of circuits that the program had to comb through grew at a furious rate as the number of input variables increased"

Combinatorial Explosion and Disillusionment

"Today, our optimism in even trying an enumerative approach may seem utterly naive [...] with an initial surge of excitement as toy problems were successfully solved, followed by disillusionment as the full seriousness of the combinatorial explosion phenomenon became apparent."

EXP

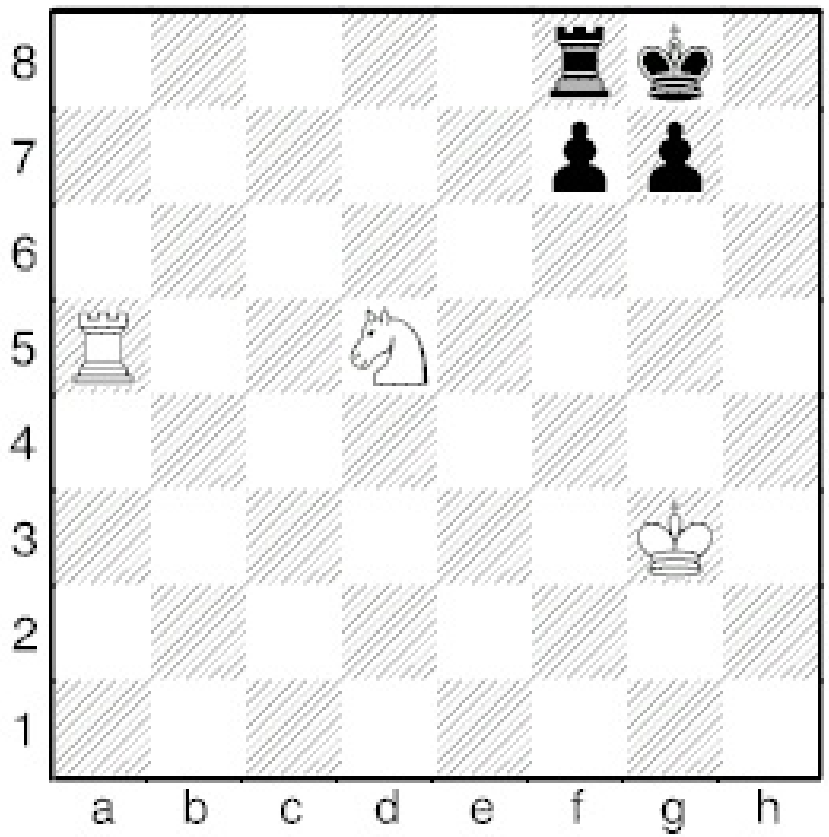
EXP is the set of problems that can be solved in exponential time ($O(2^{p(n)})$ where $p(n)$ is a polynomial).

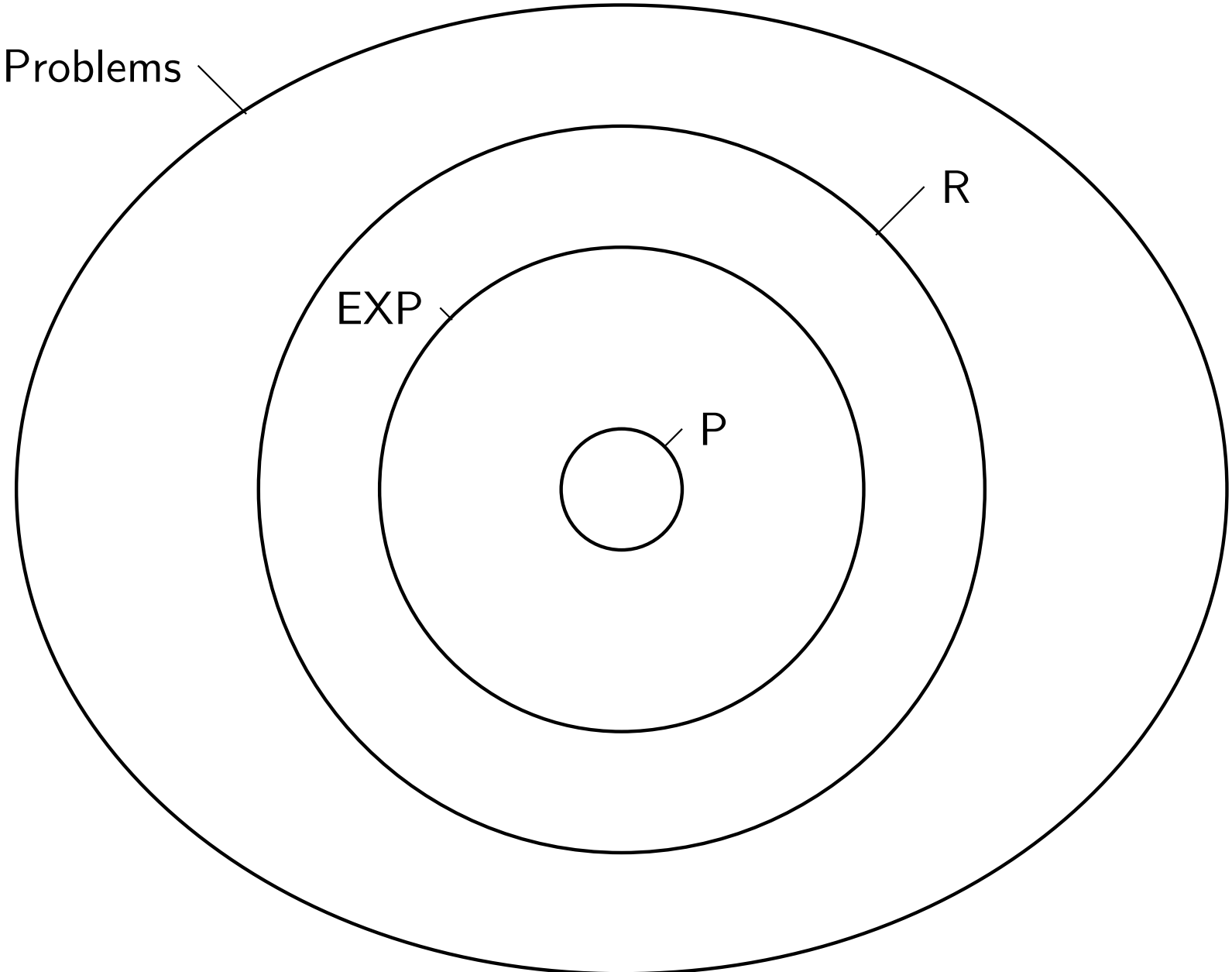
Examples

Deciding if a position in chess, checkers, or Go, is a winning position (with an infinite board).

$n \times n$ Chess

The $n \times n$ Chess problem seeks to decide whether a given configuration on a $n \times n$ chess board wins.



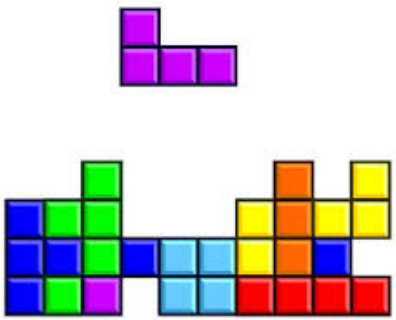

Problems
R
EXP
P

NP

NP is the set of problems whose solutions can be verified in polynomial time.

Examples

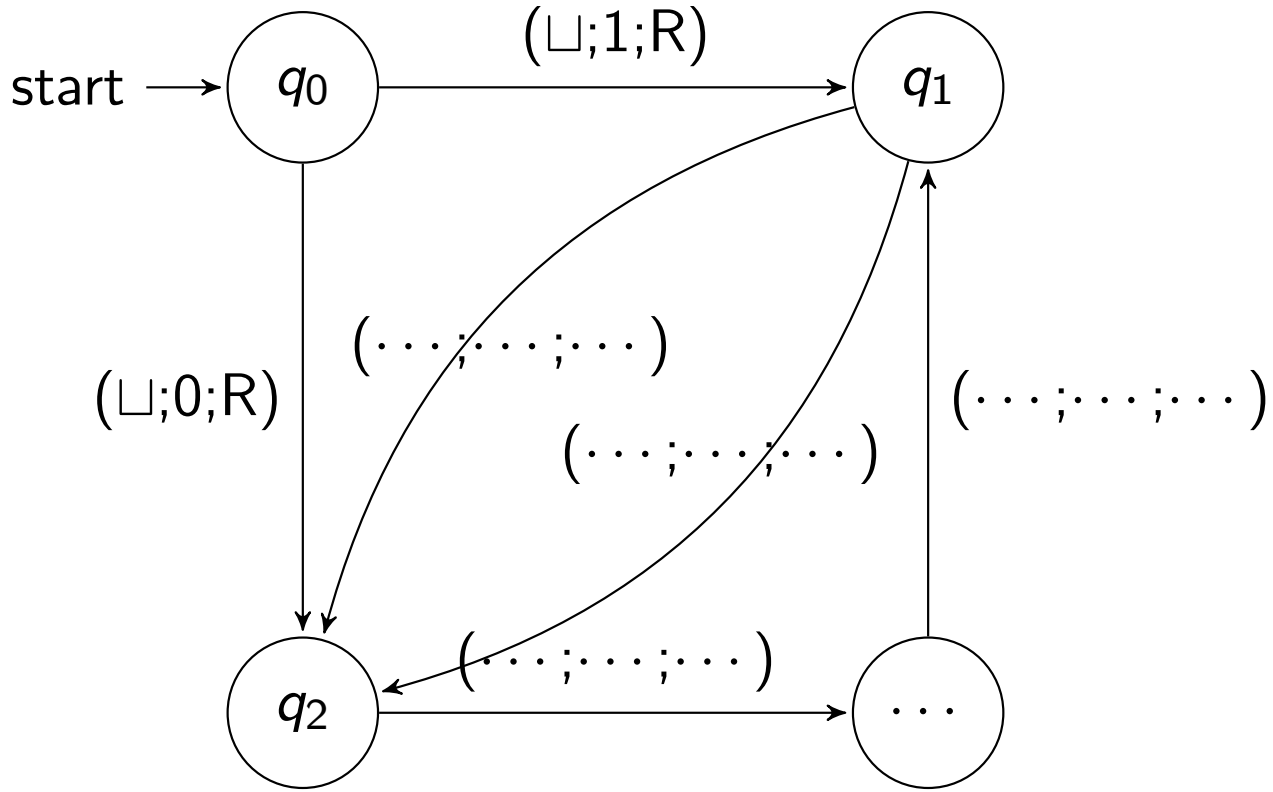
Prime factorization, minimum circuit size, satisfiability of Boolean expressions, Sudoku, Tetris, Minesweeper, Nonograms, Eternity II, Traveling Salesman Problem, etc.



Let us play Tetris (Can you survive versus can you die)

NP

NP is the set of problems that can be solved in polynomial time ($O(p(n))$ where $p(n)$ is a polynomial) by a non-deterministic Turing machine. NP stands for non-deterministic polynomial time.

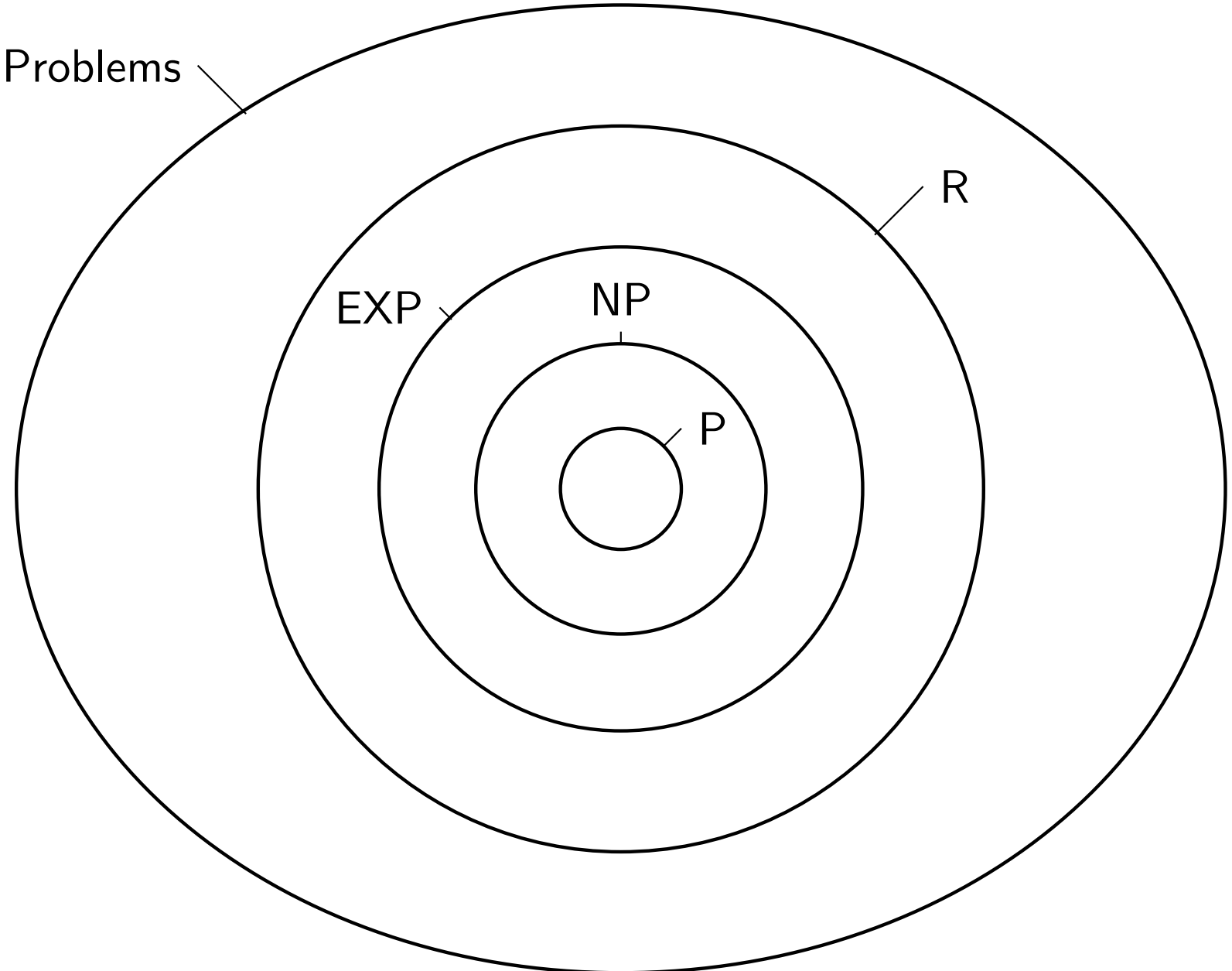


SAT

Satisfiability seeks to decide whether a given Boolean formula can be true.

p	q	r	$(p \vee q) \Rightarrow (r \oplus q)$
true	true	true	false
true	true	false	true
true	false	true	true
true	false	false	false
false	true	true	false
false	true	false	true
false	false	true	true
false	false	false	true

SAT is not known to be in P.


Problems
R
EXP
NP
P

NP-hard

A problem is NP-hard if it is at least as hard as any problem in NP.

NP-complete

A problem is NP-complete if it is in NP and NP-hard.

$SAT \in NPC$

Stephen Cook proved that SAT is NP-complete in 1971 (Cook-Levin theorem, also known as Cook's theorem).

Cook' Turing Award Citation

"For his advancement of our understanding of the complexity of computation in a significant and profound way."

Reduction

A reduction is an algorithm for transforming one problem into another problem. A many-one reduction is an algorithm for transforming one decision problem into another decision problem.

NP-complete Problems

Richard Karp, in 1972, proved the NP-completeness of 21 problems by showing a polynomial time many-one reduction of SAT to each of these problems.

Karps' Turing Award Citation

"most notably, [for his] contributions to the theory of NP-completeness. Karp introduced the now standard methodology for proving problems to be NP-complete which has led to the identification of many theoretical and practical problems as being computationally difficult."

DNF

A Boolean formula is in disjunctive normal form if and only if it is a disjunction of conjunctions of propositions and their negation.

$$(\neg p \wedge \neg q) \vee (r \wedge \neg q) \vee (q \wedge \neg r)$$

CNF

A Boolean formula is in conjunctive normal form if and only if it is a conjunction of disjunctions (called clauses) of propositions and their negation.

$$(\neg p \vee q \vee r) \wedge (\neg q \vee \neg r)$$

3-SAT

3-SAT is the special case in which a Boolean formula in CNF has at most three variables per clause.

$$(\neg p \vee q \vee r) \wedge (\neg q \vee \neg r \vee s) \wedge (\neg q \vee \neg r \vee \neg s)$$

$3SAT \in NPC$

3-SAT is NP-complete.

Proof Sketch

We rewrite any k-SAT formula into a 3-SAT one.

- $(x_1 \vee x_2)$ becomes $(x_1 \vee x_2 \vee x_{new}) \wedge (x_1 \vee x_2 \vee \neg x_{new})$
- (x_1) becomes $(x_1 \vee x_{new_1} \vee x_{new_2}) \wedge (x_1 \vee x_{new_1} \vee \neg x_{new_2}) \wedge (x_1 \vee \neg x_{new_1} \vee x_{new_2}) \wedge (x_1 \vee \neg x_{new_1} \vee \neg x_{new_2})$
- $(x_1 \vee \cdots \vee x_k)$ becomes $(x_1 \vee x_2 \vee x_{new_1}) \wedge (x_3 \vee \neg x_{new_1} \vee x_{new_2}) \vee (x_4 \vee \neg x_{new_2} \vee x_{new_3}) \wedge \ldots \wedge (x_{k-1} \vee x_k \vee \neg x_{new_{k-3}})$

This transformation is polynomial.

$$(x_1 \vee x_2 \vee x_3 \vee x_4)$$

$$(x_1 \vee x_2 \vee x_{new_1}) \wedge (x_3 \vee x_4 \vee \neg x_{new_1})$$

Now we Use 3-SAT

Now we can prove that an NP problem is NP-complete by reducing 3-SAT to it.

$2SAT \in P$

2-SAT is in P.

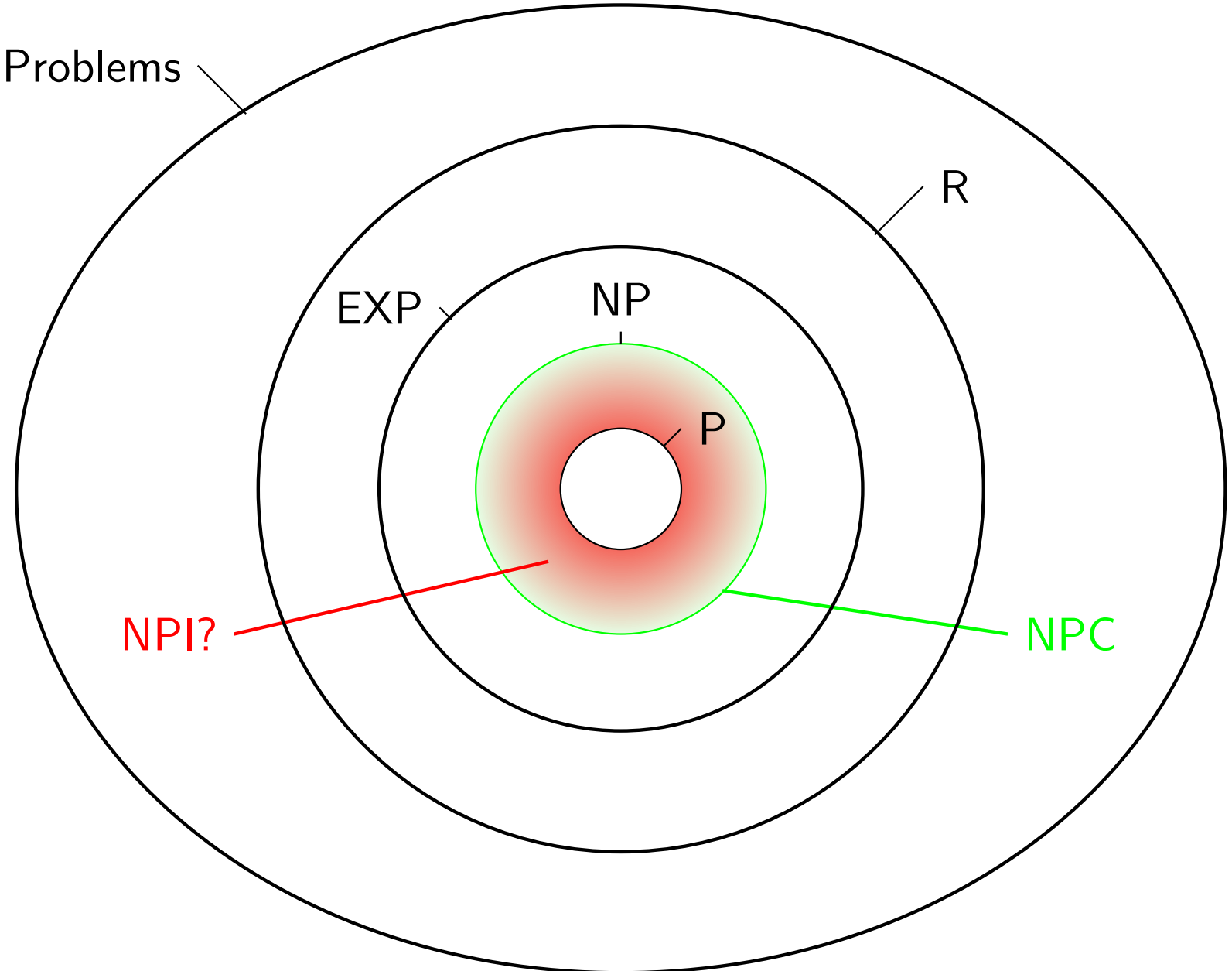
$$(\neg p \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee \neg r)$$

The Exponential Time Hypothesis

The (unproven) exponential time hypothesis, by Russell Impagliazzo and Ramamohan Paturi (1999), proposes that NP-complete problems cannot be solved in subexponential time (faster than polynomial but slower than exponential) in the worst case.

NPI

Prime factorization and minimum circuit size are candidate to be NP-intermediate, that is NP, not in P and not NP-complete.


Problems
R
EXP
NP
P
NPI?
NPC

Complexity Classes

There are 495 classes and counting in the complexity zoo. https://complexityzoo.uwaterloo.ca.

What do we Know?

- There are problems which can be solved in exponential time but cannot be solved in polynomial time.
- If $P \neq NP$ then NPI, NPC and P are three disjoint nonempty classes.
- All regular languages are solvable in polynomial time, but not vice versa; the set of all natural numbers with as many digits 1 as digits 2 as digits 3 can be decided in polynomial time but is not regular.

P=NP?

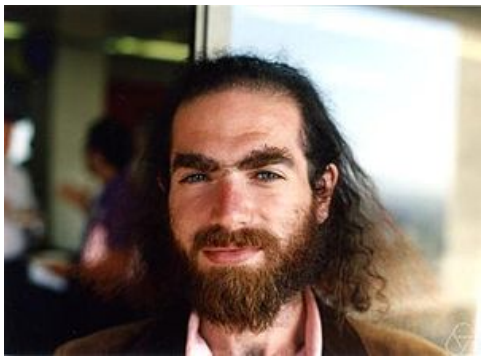
The question whether P=NP is one of the six remaining Millennium Prize Problems.

Millennium Prize

The Millennium Prize problems are seven problems proposed by the Clay Mathematics Institute in 2000. There is a one million US dollars prize for being the first one solving any one of these problems.

Stories ...

As of today, six of the Millennium Prize problems remain unsolved. The Poincaré conjecture was solved by Grigori Perelman. Grigori Perelman thought that his proof owed much to Richard Hamilton's work and declined the price. Richard Hamilton did not get Millennium Prize but he got half of the one million US dollar Shaw Prize. The Shaw Prize is a prize by the Shaw Prize Foundation. The prize is named after Sir Run Run Shaw who, with his brother, created the Shaw Organisation in Singapore.



And more Stories

"Logicomix: An Epic Search for Truth" is a comics by Apostolos Doxiad and Christos Papadimitriou. It tells the story of Cantor, Hilbert, Gödel, Turing and infinity, logic, the Entscheindung problem and many other things.

Attribution