P (complexity)

In computational complexity theory, **P**, also known as **PTIME** or **DTIME**($n^{O(1)}$), is a fundamental complexity class. It contains all decision problems that can be solved by a deterministic Turing machine using a polynomial amount of computation time, or polynomial time.

Cobham's thesis holds that P is the class of computational problems that are "efficiently solvable" or "tractable"; in practice, some problems not known to be in P have practical solutions, and some that are in P do not, but this is a useful rule of thumb.

1 Definition

A language L is in P if and only if there exists a deterministic Turing machine M, such that

- M runs for polynomial time on all inputs
- For all x in L, M outputs 1
- For all x not in L, M outputs 0

P can also be viewed as a uniform family of boolean circuits. A language L is in P if and only if there exists a polynomial-time uniform family of boolean circuits $\{C_n : n \in \mathbb{N}\}$, such that

- For all $n \in \mathbb{N}$, C_n takes n bits as input and outputs 1 bit
- For all x in L, $C_{|x|}(x) = 1$
- For all x not in L, $C_{|x|}(x) = 0$

The circuit definition can be weakened to use only a logspace uniform family without changing the complexity class.

2 Notable problems in P

P is known to contain many natural problems, including the decision versions of linear programming, calculating the greatest common divisor, and finding a maximum matching. In 2002, it was shown that the problem of determining if a number is prime is in P.[1] The related class of function problems is FP.

Several natural problems are complete for P, including *st*-connectivity (or reachability) on alternating graphs.[2] The article on P-complete problems lists further relevant problems in P.

3 Relationships to other classes

A generalization of P is NP, which is the class of decision problems decidable by a non-deterministic Turing machine that runs in polynomial time. Equivalently, it is the class of decision problems where each "yes" instance has a polynomial size certificate, and certificates can be checked by a polynomial time deterministic Turing machine. The class of problems for which this is true for the "no" instances is called co-NP. P is trivially a subset of NP and of co-NP; most experts believe it is a proper subset,[3] although this (the $\mathsf{P} \subsetneq \mathsf{NP}$ hypothesis) remains unproven. Another open problem is whether NP = co-NP (a negative answer would imply $\mathsf{P} \subsetneq \mathsf{NP}$).

P is also known to be at least as large as L, the class of problems decidable in a logarithmic amount of memory space. A decider using $O(\log n)$ space cannot use more than $2^{O(\log n)} = n^{O(1)}$ time, because this is the total number of possible configurations; thus, L is a subset of P. Another important problem is whether L = P. We do know that P = AL, the set of problems solvable in logarithmic memory by alternating Turing machines. P is also known to be no larger than PSPACE, the class of problems decidable in polynomial space. Again, whether P = PSPACE is an open problem. To summarize:

$$\mathsf{L} \subseteq \mathsf{AL} = \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXPTIME}.$$

Here, EXPTIME is the class of problems solvable in exponential time. Of all the classes shown above, only two strict containments are known:

- P is strictly contained in EXPTIME. Consequently, all EXPTIME-hard problems lie outside P, and at least one of the containments to the right of P above is strict (in fact, it is widely believed that all three are strict).
- L is strictly contained in PSPACE.

The most difficult problems in P are P-complete problems.

Another generalization of P is P/poly, or Nonuniform Polynomial-Time. If a problem is in P/poly, then it can

be solved in deterministic polynomial time provided that an advice string is given that depends only on the length of the input. Unlike for NP, however, the polynomial-time machine doesn't need to detect fraudulent advice strings; it is not a verifier. P/poly is a large class containing nearly all practical problems, including all of BPP. If it contains NP, then the polynomial hierarchy collapses to the second level. On the other hand, it also contains some impractical problems, including some undecidable problems such as the unary version of any undecidable problem.

In 1999, Jin-Yi Cai and D. Sivakumar, building on work by Mitsunori Ogihara, showed that if there exists a sparse language that is P-complete, then L = P.[4]

4 Properties

Polynomial-time algorithms are closed under composition. Intuitively, this says that if one writes a function that is polynomial-time assuming that function calls are constant-time, and if those called functions themselves require polynomial time, then the entire algorithm takes polynomial time. One consequence of this is that P is low for itself. This is also one of the main reasons that P is considered to be a machine-independent class; any machine "feature", such as random access, that can be simulated in polynomial time can simply be composed with the main polynomial-time algorithm to reduce it to a polynomial-time algorithm on a more basic machine.

Languages in P are also closed under reversal, intersection, union, concatenation, Kleene closure, inverse homomorphism, and complementation.[5]

5 Pure existence proofs of polynomial-time algorithms

Some problems are known to be solvable in polynomial-time, but no concrete algorithm is known for solving them. For example, the Robertson–Seymour theorem guarantees that there is a finite list of forbidden minors that characterizes (for example) the set of graphs that can be embedded on a torus; moreover, Robertson and Seymour showed that there is an $O(n^3)$ algorithm for determining whether a graph has a given graph as a minor. This yields a nonconstructive proof that there is a polynomial-time algorithm for determining if a given graph can be embedded on a torus, despite the fact that no concrete algorithm is known for this problem.

6 Alternative characterizations

In descriptive complexity, P can be described as the problems expressible in FO(LFP), the first-order logic with a least fixed point operator added to it, on ordered structures. In Immerman's 1999 textbook on descriptive complexity,[6] Immerman ascribes this result to Vardi[7] and to Immerman.[8]

It was published in 2001 that PTIME corresponds to (positive) range concatenation grammars.[9]

7 History

Kozen[10] states that Cobham and Edmonds are "generally credited with the invention of the notion of polynomial time." Cobham invented the class as a robust way of characterizing efficient algorithms, leading to Cobham's thesis. However, H. C. Pocklington, in a 1910 paper,[11][12] analyzed two algorithms for solving quadratic congruences, and observed that one took time "proportional to a power of the logarithm of the modulus" and contrasted this with one that took time proportional "to the modulus itself or its square root", thus explicitly drawing a distinction between an algorithm that ran in polynomial time versus one that did not.

8 Notes

[1] Manindra Agrawal, Neeraj Kayal, Nitin Saxena, "PRIMES is in P", *Annals of Mathematics* 160 (2004), no. 2, pp. 781–793.

[2] Immerman, Neil (1999). *Descriptive Complexity*. New York: Springer-Verlag. ISBN 0-387-98600-6.

[3] Johnsonbaugh, Richard; Schaefer, Marcus, *Algorithms*, 2004 Pearson Education, page 458, ISBN 0-02-360692-4

[4] Jin-Yi Cai and D. Sivakumar. Sparse hard sets for P: resolution of a conjecture of Hartmanis. *Journal of Computer and System Sciences*, volume 58, issue 2, pp.280–296. 1999. ISSN 0022-0000. At Citeseer

[5] Hopcroft, John E.; Rajeev Motwani; Jeffrey D. Ullman (2001). *Introduction to automata theory, languages, and computation* (2. ed.). Boston: Addison-Wesley. pp. 425–426. ISBN 0201441241.

[6] Immerman, Neil (1999). *Descriptive Complexity*. New York: Springer-Verlag. p. 66. ISBN 0-387-98600-6.

[7] Vardi, Moshe Y. (1982). "The Complexity of Relational Query Languages". *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*. pp. 137–146. doi:10.1145/800070.802186.

[8] Immerman, Neil (1982). "Relational Queries Computable in Polynomial Time". *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*. pp. 147–152. doi:10.1145/800070.802187. Revised version in *Information and Control*, 68 (1986), 86–104.

[9] Laura Kallmeyer (2010). *Parsing Beyond Context-Free Grammars*. Springer Science & Business Media. pp. 5 and 37. ISBN 978-3-642-14846-0. citing http://mjn.host.cs.st-andrews.ac.uk/publications/2001d.pdf for the proof

[10] Kozen, Dexter C. (2006). *Theory of Computation*. Springer. p. 4. ISBN 1-84628-297-7.

[11] Pocklington, H. C. (1910–1912). "The determination of the exponent to which a number belongs, the practical solution of certain congruences, and the law of quadratic reciprocity". *Proc. Cambridge Phil. Soc.* **16**: 1–5.

[12] Gautschi, Walter (1994). *Mathematics of computation, 1943–1993: a half-century of computational mathematics: Mathematics of Computation 50th Anniversary Symposium, August 9–13, 1993, Vancouver, British Columbia*. Providence, RI: American Mathematical Society. pp. 503–504. ISBN 0-8218-0291-7.

9 References

- Cobham, Alan (1965). "The intrinsic computational difficulty of functions". *Proc. Logic, Methodology, and Philosophy of Science II*. North Holland.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, Second Edition. MIT Press and McGraw–Hill, 2001. ISBN 0-262-03293-7. Section 34.1: Polynomial time, pp. 971–979.
- Papadimitriou, Christos H. (1994). *Computational complexity*. Reading, Mass.: Addison–Wesley. ISBN 0-201-53082-1.
- Sipser, Michael (2006). *Introduction to the Theory of Computation, 2nd Edition*. Course Technology Inc. ISBN 0-534-95097-3. Section 7.2: The Class P, pp. 256–263;.

10 External links

- *Complexity Zoo*: Class P
- *Complexity Zoo*: Class P/poly

11 Text and image sources, contributors, and licenses

11.1 Text

- **P (complexity)** *Source:* https://en.wikipedia.org/wiki/P_(complexity)?oldid=777460910 *Contributors:* Stevertigo, Michael Hardy, Dcoetzee, Tea2min, Giftlite, Markus Krötzsch, Wmahan, Neilc, Gdr, Creidieki, DcoetzeeBot~enwiki, EmilJ, Oliphaunt, Adiel, Marudubshinki, Rjwilmsi, OmriSegal, GünniX, Intgr, Chobot, Visor, Bgwhite, YurikBot, Trovatore, Esqg, Ott2, Cedar101, That Guy, From That Show!, SmackBot, Gelingvistoj, Perlmonger42, Ylloh, CRGreathouse, NotQuiteEXPComplete, TOGASHI Jin, Thijs!bot, Electron9, GromXXVII, Illuminatedwax, Yonidebot, Derlay, Stokkink, Adam Zivner, TXiKiBoT, Jobu0101, PaulTanenbaum, SolifyDolphin~enwiki, AlanUS, ClueBot, Kotniski, Kraven007mega, Mikaey, Uzdzislaw, Addbot, DOI bot, SpellingBot, MrVanBot, CarsracBot, Favonian, مانع, Yobot, KamikazeBot, Rubinbot, Materialscientist, Citation bot, Twri, LucienBOT, Citation bot 1, RedBot, RobinK, RjwilmsiBot, Ripchip Bot, Timde, Carbo1200, Dcirovic, ZéroBot, Luke18:2-8, ClueBot NG, Jeremymy, Max Longint, BattyBot, David.moreno72, Deltahedron, Jrgauthier, Comp.arch, SJ Defender, Monkbot, IagoQnsi, JMP EAX, Qzd, BarryFruitman, Justeditingtoday and Anonymous: 49

11.2 Images

11.3 Content license