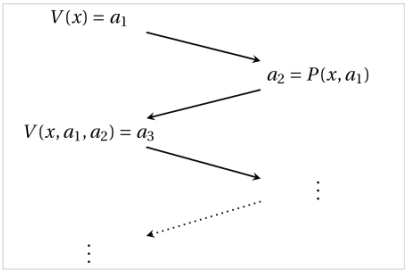
Интерактивная система проверки

В теории вычислительной сложности **система интерактивного доказательства** — это абстрактная машина, которая моделирует вычисления как обмен сообщениями между двумя сторонами: *доказывающей* и *проверяющей*. Стороны взаимодействуют, обмениваясь сообщениями, чтобы определить, принадлежит ли заданная строка языку или нет. Предполагается, что доказывающая сторона обладает неограниченными вычислительными ресурсами, но ей нельзя доверять, в то время как проверяющая сторона имеет ограниченную вычислительную мощность, но считается всегда честной. Сообщения передаются между проверяющим и доказывающим до тех пор, пока проверяющий не получит ответ на вопрос и не «убедится» в его правильности.



Общее представление об интерактивном протоколе доказательства.

Все интерактивные системы проверки знаний предъявляют два требования:

- **Полнота**: если утверждение верно, то честный доказывающий (то есть тот, кто правильно следует протоколу) может убедить честного проверяющего в том, что утверждение действительно верно.
- **Достоверность**: если утверждение ложно, то ни один доказывающий, даже если он не следует протоколу, не сможет убедить честного проверяющего в том, что утверждение верно, за исключением небольшой вероятности.

Специфика системы и, следовательно, класс сложности языков, которые она может распознавать, зависят от того, какие ограничения накладываются на верификатор, а также от его возможностей. Например, большинство систем интерактивного доказательства критически зависят от способности верификатора делать случайный выбор. Это также зависит от характера передаваемых сообщений — их количества и содержания. Было установлено, что системы интерактивного доказательства имеют важное значение для традиционных классов сложности, определяемых с использованием только одной машины. Основными классами сложности, описывающими системы интерактивного доказательства, являются **AM** и **IP**.

Предыстория

Каждая система интерактивного доказательства определяет формальный язык строк L. **Адекватность** системы доказательства означает, что ни один доказывающий не может заставить проверяющего принять ложное утверждение $y \notin L$ с вероятностью выше некоторой малой величины. Верхняя граница этой вероятности называется **ошибкой адекватности** системы доказательства. Более формально, для каждого доказывающего $(\tilde{\mathcal{P}})$ и каждого $y \notin L$:

$$\Pr[(\perp, (\text{accept})) \leftarrow (\tilde{\mathcal{P}})(y) \leftrightarrow (\mathcal{V})(y)] < \epsilon.$$

для некоторых $\epsilon \ll 1$. Пока ошибка корректности ограничена полиномиальной долей от потенциального времени работы верификатора (т. е. $\epsilon \leq 1/\text{poly}(|y|)$), всегда можно повысить корректность до тех пор, пока ошибка корректности не станет пренебрежимо малой функцией относительно времени работы верификатора. Это достигается путем повторения доказательства и принятия только в том случае, если все доказательства корректны. После ℓ повторений ошибка корректности ϵ будет снижена до ϵ^{ℓ}.[1]

Классы интерактивных доказательств

NP

Класс сложности NP можно рассматривать как очень простую систему доказательств. В этой системе проверяющая сторона — это детерминированная машина, работающая за полиномиальное время (машина P). Протокол выглядит следующим образом:

- Доказывающая сторона анализирует входные данные, вычисляет решение, используя свои неограниченные возможности, и возвращает сертификат доказательства полиномиального размера.
- Проверяющий модуль проверяет действительность сертификата за детерминированное полиномиальное время. Если сертификат действителен, модуль принимает его, в противном случае — отклоняет.

В случае наличия действительного сертификата подтверждения доказывающий всегда может убедить проверяющего, предоставив ему этот сертификат. Однако в случае отсутствия действительного сертификата подтверждения входные данные не относятся к языку, и никакой доказывающий, каким бы злонамеренным он ни был, не сможет убедить проверяющего в обратном, поскольку любой сертификат подтверждения будет отклонён.

Протоколы Артура — Мерлина и Мерлина — Артура

Хотя NP можно рассматривать как использующее взаимодействие, только в 1985 году две независимые группы исследователей сформулировали концепцию вычислений через взаимодействие (в контексте теории сложности). Один из подходов, предложенный Ласло Бабаи, опубликовавшим статью «Обмен теории групп на случайность»[2], определил иерархию классов *Артура — Мерлина* (**AM**). В этой презентации Артур (верификатор) — это вероятностная машина с полиномиальным временем работы, а Мерлин (доказывающий) обладает неограниченными ресурсами.

Класс **MA**, в частности, представляет собой простое обобщение описанного выше взаимодействия NP, в котором проверяющая сторона является вероятностной, а не детерминированной. Кроме того, проверяющая сторона не обязана всегда принимать действительные сертификаты и отклонять недействительные, она может быть более снисходительной:

- **Полнота:** если строка принадлежит языку, то доказывающий должен быть в состоянии предоставить сертификат, который проверяющий примет с вероятностью не менее 2/3 (в зависимости от случайного выбора проверяющего).
- **Звучность:** если строка не принадлежит языку, то ни один злоумышленник не сможет убедить проверяющего принять строку с вероятностью, превышающей 1/3.

Эта машина потенциально более мощная, чем обычный NP протокол взаимодействия, а сертификаты не менее удобны для проверки, поскольку **BPP** алгоритмы считаются абстрактными практическими вычислениями (см. BPP).

Протокол публичной монеты в сравнении с протоколом частной монеты

В протоколе *публичной монеты* случайные выборки, сделанные верификатором, становятся общедоступными. В протоколе приватной монеты они остаются конфиденциальными.

На той же конференции, где Бабай определил свою систему доказательств для **MA**, Шафи Голдвассер, Сильвио Микали и Чарльз Ракофф[3] опубликовали статью, определяющую систему интерактивных доказательств **IP**[$f(n)$]. В нем используются те же машины, что и в протоколе **MA**, за исключением того, что для ввода размера n *раундов* допускается $f(n)$. В каждом раунде проверяющий выполняет вычисления и передаёт сообщение доказывающему, а доказывающий выполняет вычисления и передаёт информацию обратно проверяющему. В конце проверяющий должен принять решение. Например, в протоколе **IP**[3] последовательность будет такой: VPVPVPV, где V — ход проверяющего, а P — ход доказывающего.

В протоколах Артура — Мерлина Бабаи определил аналогичный класс **AM**[$f(n)$], который допускает $f(n)$ раундов, но накладывает на машину дополнительное условие: проверяющий должен показать доказывающему все случайные биты, которые он использует в своих вычислениях. В результате проверяющий не может ничего «скрыть» от доказывающего, потому что доказывающий достаточно силён, чтобы смоделировать всё, что делает проверяющий, если он знает, какие случайные биты тот использовал. Это называется протоколом *публичной монеты*, потому что случайные биты («подбрасывание монеты») видны обеим машинам. Подход **IP** напротив, называется протоколом *частной монеты*.

The essential problem with public coins is that if the prover wishes to maliciously convince the verifier to accept a string which is not in the language, it seems like the verifier might be able to thwart its plans if it can hide its internal state from it. This was a primary motivation in defining the **IP** proof systems.

In 1986, Goldwasser and Sipser[4] showed, perhaps surprisingly, that the verifier's ability to hide coin flips from the prover does it little good after all, in that an Arthur–Merlin public coin protocol with only two more rounds can recognize all the same languages. The result is that public-coin and private-coin protocols are roughly equivalent. In fact, as Babai shows in 1988, **AM**[k]=**AM** for all constant k, so the **IP**[k] have no advantage over **AM**.[5]

To demonstrate the power of these classes, consider the graph isomorphism problem, the problem of determining whether it is possible to permute the vertices of one graph so that it is identical to another graph. This problem is in **NP**, since the proof certificate is the permutation which makes the graphs equal. It turns out that the complement of the graph isomorphism problem, a co-**NP** problem not known to be in **NP**, has an **AM** algorithm and the best way to see it is via a private coins algorithm.[6]

IP

Private coins may not be helpful, but more rounds of interaction are helpful. If we allow the probabilistic verifier machine and the all-powerful prover to interact for a polynomial number of rounds, we get the class of problems called **IP**. In 1992, Adi Shamir revealed in one of the central results of complexity theory that **IP** equals **PSPACE**, the class of problems solvable by an ordinary deterministic Turing machine in polynomial space.[7]

QIP

If we allow the elements of the system to use quantum computation, the system is called a **quantum interactive proof system**, and the corresponding complexity class is called **QIP**.[8] A series of results culminated in a 2010 breakthrough that **QIP** = **PSPACE**.[9][10]

Zero knowledge

Not only can interactive proof systems solve problems not believed to be in **NP**, but under assumptions about the existence of one-way functions, a prover can convince the verifier of the solution without ever giving the verifier information about the solution. This is important when the verifier cannot be trusted with the full solution. At first it seems impossible that the verifier could be convinced that there is a solution when the verifier has not seen a certificate, but such proofs, known as zero-knowledge proofs are in fact believed to exist for all problems in **NP** and are valuable in cryptography. Zero-knowledge proofs were first mentioned in the original 1985 paper on **IP** by Goldwasser, Micali and Rackoff for specific number theoretic languages. The extent of their power was however shown by Oded Goldreich, Silvio Micali and Avi Wigderson.[6] for all of **NP**, and this was first extended by Russell Impagliazzo and Moti Yung to all **IP**.[11]

MIP

One goal of **IP**'s designers was to create the most powerful possible interactive proof system, and at first it seems like it cannot be made more powerful without making the verifier more powerful and so impractical. Goldwasser et al. overcame this in their 1988 "Multi prover interactive proofs: How to remove intractability assumptions", which defines a variant of **IP** called **MIP** in which there are *two* independent provers.[12] The two provers cannot communicate once the verifier has begun sending messages to them. Just as it's easier to tell if a criminal is lying if he and his partner are interrogated in separate rooms, it's considerably easier to detect a malicious prover trying to trick the verifier into accepting a string not in the language if there is another prover it can double-check with.

In fact, this is so helpful that Babai, Fortnow, and Lund were able to show that **MIP** = **NEXPTIME**, the class of all problems solvable by a nondeterministic machine in *exponential time*, a very large class.[13] NEXPTIME contains PSPACE, and is believed to strictly contain PSPACE. Adding a constant number of additional provers beyond two does not enable recognition of any more languages. This result paved the way for the celebrated PCP theorem, which can be considered to be a "scaled-down" version of this theorem.

MIP also has the helpful property that zero-knowledge proofs for every language in **NP** can be described without the assumption of one-way functions that **IP** must make. This has bearing on the design of provably unbreakable cryptographic algorithms.[12] Moreover, a **MIP** protocol can recognize all languages in **IP** in only a constant number of rounds, and if a third prover is added, it can recognize all languages in **NEXPTIME** in a constant number of rounds, showing again its power over **IP**.

Известно, что для любой константы k система MIP с k проверяющими и полиномиальным количеством раундов может быть преобразована в эквивалентную систему всего с двумя проверяющими и постоянным количеством раундов.[14]

PCP

В то время как разработчики **IP** рассматривали возможность обобщения систем интерактивного доказательства Бабаи, другие рассматривали возможность их ограничения. Очень полезной системой интерактивного доказательства является **PCP**($f(n)$, $g(n)$), которая представляет собой ограничение **MA**, где Артур может использовать только $f(n)$ случайных битов и может проверять только $g(n)$ битов сертификата доказательства, отправленного Мерлином (по сути, используя случайный доступ).

Существует ряд легко доказуемых результатов о различных классах **PCP**. $\mathbf{PCP}(0, \mathrm{poly})$, класс машин с полиномиальным временем работы, не использующих случайность, но имеющих доступ к сертификату, — это просто **NP**. $\mathbf{PCP}(\mathrm{poly}, 0)$, класс машин с полиномиальным временем работы, имеющих доступ к полиномиальному количеству случайных битов, — это **co-RP**. Первым важным результатом работы Арори и Сафры стало то, что $\mathbf{PCP}(\log, \log) = \mathbf{NP}$; другими словами, если проверяющий в протоколе **NP** может выбирать только $O(\log n)$ битов из сертификата доказательства, это не будет иметь никакого значения, если у него есть $O(\log n)$ случайных битов.[15]

Кроме того, теорема PCP утверждает, что количество обращений к доказательству может быть сведено к константе. То есть $\mathbf{NP} = \mathbf{PCP}(\log, O(1))$.[16] Они использовали эту ценную характеристику **NP**, чтобы доказать, что приближённые алгоритмы не существуют для оптимизационных версий некоторых NP-полных задач, если только P = NP. Такие задачи сейчас изучаются в области, известной как сложность аппроксимации.

Смотрите также

- Машина Оракула
- Доказательство знания

Ссылки

1. Голдрейх, Одед (2002), *Криптография с нулевым разглашением спустя двадцать лет после её изобретения*, ECCC TR02-063 (https://eccc.weizmann.ac.il/report/2002/063/).
2. Ласло Бабай. Использование теории групп для получения случайных результатов (http://portal.acm.org/citation.cfm?id=22192). *Материалы семнадцатого ежегодного симпозиума по теории вычислений*, ACM. 1985.
3. АрхивированоРасширенная аннотация (http://crypto.cs.mcgill.ca/~crepeau/COMP647/2007/TOPIC02/GMR89.pdf) Голдвассер, С.; Микали, С.; Ракофф, К. (1989). "Сложность интерактивных систем доказательства" *(PDF)*. **Журнал SIAM по вычислительной технике**. 18 (1): 186–208. doi (https://doi.org/10.1137%2F0218012):10.1137/0218012 . (https://search.worldcat.org/issn/1095-7111)ISSN 1095-7111 (http://theory.lcs.mit.edu/~cis/pubs/shafi/1985-stoc.pdf) . (https://web.archive.org/web/20060623020231/http://theory.lcs.mit.edu/~cis/pubs/shafi/1985-stoc.pdf) 23 июня 2006 г. в Wayback Machine
4. Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems (http://theory.lcs.mit.edu/~cis/pubs/shafi/1986-stoc.pdf) Archived (https://web.archive.org/web/20050127045423/http://theory.lcs.mit.edu/~cis/pubs/shafi/1986-stoc.pdf) 2005-01-27 at the Wayback Machine. *Proceedings of ACM STOC'86*, pp. 58–68. 1986.
5. László Babai and Shlomo Moran. Arthur–Merlin games: a randomized proof system, and a hierarchy of complexity classes (http://portal.acm.org/citation.cfm?id=49987). *Journal of Computer and System Sciences*, 36: p.254–276. 1988.
6. O. Goldreich, S. Micali, A. Wigderson. Proofs that yield nothing but their validity (http://portal.acm.org/citation.cfm?id=116852). *Journal of the ACM*, volume 38, issue 3, p.690–728. July 1991.
7. Adi Shamir. IP = PSPACE (http://portal.acm.org/citation.cfm?doid=146585.146609). *Journal of the ACM*, volume 39, issue 4, p.869–877. October 1992.
8. Tsuyoshi Ito; Hirotada Kobayashi; John Watrous (2010). "Quantum interactive proofs with weak error bounds". arXiv:1012.4427v2 (https://arxiv.org/abs/1012.4427v2) [quant-ph (https://arxiv.org/archive/quant-ph)].
9. Джайн, Рахул; Цзи, Чжэнфэн; Упадхьяй, Сарвагья; Уотрус, Джон (2010). "QIP = PSPACE". *STOC '10: Материалы 42-го симпозиума ACM по теории вычислений*. ACM. стр. 573–582. ISBN 978-1-4503-0050-6.
10. Аароносн, С. (2010). «Прорыв в области QIP = PSPACE». *Communications of the ACM*. **53** (12): 101. doi:10.1145/1859204.1859230 (https://doi.org/10.1145%2F1859204.1859230). S2CID 34380788 (https://api.semanticscholar.org/CorpusID:34380788).
11. Рассел Импальяццо, Моти Юнг: Прямые вычисления с минимальным разглашением. CRYPTO 1987: 40–51 [1] (https://link.springer.com/chapter/10.1007%2F3-540-48184-2_4)
12. М. Бен-Ор, Шафи Голдвассер, Дж. Килиан и А. Вигдерсон. Интерактивные доказательства с несколькими проверяющими: Как избавиться от предположений о неразрешимости (http://theory.lcs.mit.edu/~cis/pubs/shafi/1988-stoc-bgkw.pdf). *Материалы 20-го симпозиума ACM по теории вычислений*, стр. 113–121. 1988.
13. László Babai; L. Fortnow; C. Lund (1991). "Non-deterministic exponential time has two-prover interactive protocols. Computational Complexity" (https://web.archive.org/web/20070208015711/https://citeseer.ist.psu.edu/15039.html). pp. 3–40. Archived from the original (http://citeseer.ist.psu.edu/15039.html) on 8 February 2007.
14. Ben-Or, Michael; Goldwasser, Shafi; Kilian, Joe; Widgerson, Avi (1988). "Multi-prover interactive proofs: How to remove intractability" (https://web.archive.org/web/20100713035025/http://groups.csail.mit.edu/cis/pubs/shafi/1988-stoc-bgkw.pdf) (PDF). *Proceedings of the twentieth annual ACM symposium on Theory of computing - STOC '88*. pp. 113–131. doi:10.1145/62212.62223 (https://doi.org/10.1145%2F62212.62223). ISBN 0897912640. S2CID 11008365 (https://api.semanticscholar.org/CorpusID:11008365). Archived from the original (http://groups.csail.mit.edu/cis/pubs/shafi/1988-stoc-bgkw.pdf) (PDF) on 13 July 2010. Retrieved 17 November 2022.
15. Sanjeev Arora and Shmuel Safra. Probabilistic Checking of Proofs: A New Characterization of NP (http://citeseer.ist.psu.edu/arora92probabilistic.html). *Journal of the ACM*, volume 45, issue 1, pp. 70–122. January 1998.
16. Sanjeev Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and the Hardness of Approximation Problems (http://citeseer.ist.psu.edu/376426.html). Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science, pp. 13–22. 1992.

Textbooks

- Arora, Sanjeev; Barak, Boaz, "Complexity Theory: A Modern Approach" (http://www.cs.princeton.edu/theory/complexity/), Cambridge University Press, March 2009.
- Michael Sipser (1997). *Introduction to the Theory of Computation* (https://archive.org/details/introductiontoth00sips). PWS Publishing. ISBN 978-0-534-94728-6. Section 10.4: Interactive Proof Systems, pp. 354–366.
- Christos Papadimitriou (1993). *Computational Complexity* (1st ed.). Addison Wesley. ISBN 978-0-201-53082-7. Section 19.2: Games against nature and interactive protocols, pp. 469–480.

External links

- Dexter Kozen. Interactive Proofs (https://web.archive.org/web/20050224021847/http://www.cs.cornell.edu/Courses/cs682/2004sp/Lectures/l15-ip.pdf). CS682 Spring 2004 lecture notes. Department of Computer Science, Cornell University.
- Complexity Zoo:
  - MA (https://web.archive.org/web/20100727022118/http://qwiki.stanford.edu/wiki/Complexity_Zoo:M#ma), MA' (https://web.archive.org/web/20100727022118/http://qwiki.stanford.edu/wiki/Complexity_Zoo:M#maprime), MAEXP (https://web.archive.org/web/20100727022118/http://qwiki.stanford.edu/wiki/Complexity_Zoo:M#maexp), MAE (https://web.archive.org/web/20100727022118/http://qwiki.stanford.edu/wiki/Complexity_Zoo:M#mae)
  - AM (https://web.archive.org/web/20100727013744/http://qwiki.stanford.edu/wiki/Complexity_Zoo%3AA#am), AMEXP (https://web.archive.org/web/20100727013744/http://qwiki.stanford.edu/wiki/Complexity_Zoo%3AA#amexp), AM intersect co-AM (https://web.archive.org/web/20100727013744/http://qwiki.stanford.edu/wiki/Complexity_Zoo%3AA#amicoam), AM[polylog] (https://web.archive.org/web/20100727013744/http://qwiki.stanford.edu/wiki/Complexity_Zoo%3AA#ampolylog), coAM (http://qwiki.caltech.edu/wiki/Complexity_Zoo:C#coam), BP•NP (https://web.archive.org/web/20100727033302/http://qwiki.stanford.edu/wiki/Complexity_Zoo%3AB#bpnp)
  - QMA (https://web.archive.org/web/20100727032326/http://qwiki.stanford.edu/wiki/Complexity_Zoo:Q#qma), QMA+ (https://web.archive.org/web/20100727032326/http://qwiki.stanford.edu/wiki/Complexity_Zoo:Q#qma+), QMA(2) (https://web.archive.org/web/20100727032326/http://qwiki.stanford.edu/wiki/Complexity_Zoo:Q#qma2), QMA_{log} (https://web.archive.org/web/20100727032326/http://qwiki.stanford.edu/wiki/Complexity_Zoo:Q#qmalog), QMAM (https://web.archive.org/web/20100727032326/http://qwiki.stanford.edu/wiki/Complexity_Zoo:Q#qmam)
  - IP (https://web.archive.org/web/20100727062035/http://qwiki.stanford.edu/wiki/Complexity_Zoo:I#ip), MIP (https://web.archive.org/web/20100727022118/http://qwiki.stanford.edu/wiki/Complexity_Zoo:M#mip), IPP (https://web.archive.org/web/20100727062035/http://qwiki.stanford.edu/wiki/Complexity_Zoo:I#ipp), QIP (https://web.archive.org/web/20100727032326/http://qwiki.stanford.edu/wiki/Complexity_Zoo:Q#qip), QIP(2) (https://web.archive.org/web/20100727032326/http://qwiki.stanford.edu/wiki/Complexity_Zoo:Q#qip2), compIP (https://web.archive.org/web/20081023050027/http://qwiki.stanford.edu/wiki/Complexity_Zoo:C#compip), frIP (https://web.archive.org/web/20100727052636/http://qwiki.stanford.edu/wiki/Complexity_Zoo:F#frip)
  - PCP(r(n),q(n)) (https://web.archive.org/web/20081204124245/http://qwiki.stanford.edu/wiki/Complexity_Zoo:P#pcp)
- Larry Gonick. "Proof Positive? (https://web.archive.org/web/20120722015436/http://theory.cs.uchicago.edu/merlin/)". A comic strip about interactive proof systems.