Angewandtes Quantencomputing

Jörg Lässig¹

Online publiziert: 23. September 2020
© Der/die Autor(en) 2020

Zusammenfassung

Quantencomputing ist als Konzept mittlerweile zwar mehrere Jahrzehnte alt und hat von der Idee bis zu den heutigen Noisy Intermediate-Scale Quantum Maschinen verschiedene Phasen stetiger Entwicklung durchlaufen, jedoch insbesondere in den letzten Jahren deutlich an Dynamik gewonnen. Im aktuellen Stadium stehen Quantenrechner zur Verfügung, die zwar noch immer stark limitiert sind, jedoch die Ausführung maßgeschneiderter Algorithmen unterstützen, die bestehende Schwächen dieser Maschinen mit ihrem speziellen Design zu umgehen versuchen. Diese Algorithmen adressieren dabei mit Maschinellen Lernen, kombinatorischer Optimierung und Simulationsanwendungen sowie weiteren potenziellen Anwendungsfeldern Problemstellungen, die praktisch vielfältig relevant und deshalb von breitem, allgemeinen Interesse sind. Das prinzipielle Potenzial des Quantencomputers, in bestimmten Anwendungsfeldern deutlich leistungsfähiger zu sein als klassische Rechner, ruft viel Aufmerksamkeit hervor. Außerdem stellt bereits eine ganze Reihe von prinzipiell für jedermann zugänglichen Softwareframeworks Implementierungen aktueller Quantenalgorithmen zum Test und zur Evaluierung bereit. Der Artikel stellt die bisherige Entwicklung, aktuelle Verfahren sowie weitere Perspektiven der Technologie in Grundzügen dar und informiert über mögliche Anwendungsgebiete aber auch bekannte Grenzen des Quantencomputing-Paradigmas.

Quantencomputing verspricht im Zuge der sogenannten zweiten Quantenrevolution nichts Geringeres als die Etablierung eines ganz neuen Ansatzes, Berechnungen durchzuführen. Dieser soll den Vorstoß in neue Anwendungsgebiete der Informatik erlauben, umgesetzt durch Computer, die auf den Gesetzen der Quantenmechanik beruhen. Zum Einsatz kommen dabei Algorithmen, die mit ihrer Leistungsfähigkeit Problemstellungen adressieren, deren praktische Lösbarkeit bisher außer Reichweite lag. Diese Idee ist nicht neu, im Informatik-Spektrum haben wir bereits vor 20 Jahren, im Oktober 2000, zum Thema Quantencomputing berichtet [1]. Wo steht die Technologie „Quantencomputing“ heute, wie werden die Potenziale 20 Jahre später eingeschätzt, wie steht es um die Entwicklung bei der praktischen Umsetzung und potenziellen Anwendungen?

Eine kurze Geschichte des Quantencomputings

Nach wegbereitenden Resultaten in den 1960er- und 1970er-Jahren existiert die explizite Formulierung des Konzepts eines Quantencomputers seit den 1980er-Jahren. Es geht zurück auf Ideen von Paul Benioff, Yuri Manin und Richard Feynman. Eine wichtige Motivation war damals wie heute das Potenzial, Simulationen quantenmechanischer physikalischer Systeme gegenüber „klassischen Computern“ zu beschleunigen. Nach einigen eher akademischen Problemen, deren theoretische Lösung, beginnend mit dem Quantencomputing-Paradigma, mit dem Algorithmus von Deutsch demonstriert wurde, fand die Thematik in den 1990er-Jahren mit den Arbeiten von Peter Shor [2] und Lov Grover [3] große Beachtung. Mit Quantenalgorithmen für das Faktorisierungsproblem und die sogenannte unstrukturierte Datenbanksuche gab es nun zwei Beispiele für Verfahren, die einer Lösung mit klassischen Computern überlegen sind und einen exponentiellen bzw. polynomiellen Speed-up gegenüber etablierten Verfahren bieten. Darüber hinaus stellen sie für erstere der genannten Problemstellungen die Sicherheit der Public-Key-Infrastruktur in Frage und riefen damit sogar Geheimdienste auf den Plan.

✉ Jörg Lässig
joerg.laessig@iosb-ast.fraunhofer.de

¹ Fraunhofer IOSB, Hochschule Zittau/Görlitz, Görlitz, Deutschland

Seither ist der Quantenalgorithmen-Zoo [4] bereits in den 2000er-Jahren erheblich angewachsen und adressiert eine ganze Reihe von wichtigen Problemtypen und Problemstellungen. Im Kern vieler Algorithmen stecken dabei die Ideen der Pioniere in der einen oder anderen Form – etwa die Quanten-Fourier-Transformation. Eine hervorragende Übersicht zu diesen nun quasi schon „klassischen“ Themen des Quantencomputings und der Quanteninformatik findet man etwa im Standardwerk von Michael A. Nielsen und Isaac L. Chuang [5]. Die meisten dieser Verfahren haben gemein, dass sie einen universellen und fehlerkorrigierten Quantencomputer benötigen, um praktisch zu skalieren bzw. zu funktionieren. Damit bleiben sie bis heute weitestgehend theoretische Konzepte, deren skalierbare Implementierung auch gegenwärtig außer Reichweite liegt.

Etwa das Jahr 2010 markiert ein Umdenken hin zu praktischeren Problemstellungen, die insbesondere in den Bereichen Optimierung und maschinelles Lernen, aber auch bei der bereits genannten Simulation physikalischer Systeme zu finden sind. Ebenfalls in diese Zeit fallen – nach ersten Implementierungen um die Jahrtausendwende – zunehmende Erfolge bei der praktischen Entwicklung von Quantenrechnern, beginnend mit Quantenannealern als Implementierungen des Adiabatischen Quantencomputing-Modells [6], und ab etwa 2015 zunehmend durch Implementierungen des immer stärker fokussierten Gate-Modells [7]. Dabei wurden in den letzten Jahren signifikante Fortschritte erzielt und heute konkurrieren verschiedene Hardwareansätze wie etwa die Implementierung einer Quantencomputing-Architektur auf der Basis von Halbleitern oder Ionenfallen. Welches Konzept sich am Ende als das tragfähigste erweist, ist längst nicht ausgemacht.

Quantencomputing heute – Noisy Intermediate-Scale Quantum

Die heute oft genannten Rechner, vertreten etwa mit dem IBM Q System, der Implementierung von Rigetti oder auch dem Sycamore-Prozessor von Google, sind Gate-Modellimplementierungen auf der Basis von Halbleitern. Honeywell hingegen entwickelt Hardware auf der Basis von Ionenfallen. Im Jahr 2018 prägte John Preskill für die mit diesen Maschinen andauernde Phase den Begriff NISQ(Noisy Intermediate-Scale Quantum)-Ära des Quantencomputings [8]. Er bezeichnet damit Quantenrechner mit 50–100 Qubits und nur begrenzten Möglichkeiten zur Fehlerkorrektur. Diese sind Stand der Technik und erlauben eine Ausführung beliebig tiefer Quantenschaltkreise noch nicht, können aber für bestimmte Aufgaben die Leistung klassischer Computer bereits übertreffen – wie jener für die Quantensupremacy-Demonstration im Oktober 2019 eingesetzte 53-Qubit Prozessor von Google [9].

Eine praktisch wenig relevante Aufgabe konnte dabei in 200 s mit dem genannten Sycamore-Chip gelöst werden, anstelle – je nach Lesart in 10.000 Jahren (Aussage Google) oder 2,5 Tagen (Aussage IBM) – bei Ausführung auf dem zu dieser Zeit leistungsfähigsten klassischen Supercomputer Summit von IBM. Dies markiert also immerhin einen Speed-up um den Faktor 1080 bis 1,578 Mrd. gegenüber dieser Maschine.

Die NISQ-Ära hat mit Innovationen bei der Quantenhardware auch neue algorithmische Ansätze motiviert. Verstärkte Anstrengungen hierbei haben Verfahren des überwachten Lernens und der kombinatorischen Optimierung für diese Maschinen hervorgebracht, die Entwicklung verschiedener Frameworks zur Implementierung dieser Algorithmen schreitet zügig voran und erste anwendungsorientierte Fachbücher zur Thematik (vgl. z. B. Maria Schuld und Francesco Petruccione zum überwachten Lernen [10]) sind heute verfügbar.

Hybride Algorithmen – Simulation, maschinelles Lernen, Optimierung

Die heutigen Algorithmen ordnen sich zumeist in das Schema der sogenannten hybriden quantenklassischen Algorithmen bzw. variierbaren Schaltkreise ein [11]. Das Konzept berücksichtigt integral die Grenzen heutiger Hardware und ist in diesem Sinne intuitiv: Ein nicht allzu tief geschichteter parametrierter Quantenschaltkreis wird durch einen klassischen Rechner unter Variation der Parameter des Quantenschaltkreises iterativ mehrfach ausgeführt. Der klassische Anteil des Algorithmus optimiert die Ausgabe des Quantenschaltkreises dabei bezüglich einer Zielfunktion – etwa mittels Gradientenabstieg oder unter Anwendung anderer klassischer Optimierungsverfahren (vgl. Abb. 1).

Nach diesem Prinzip arbeiten die derzeit vielversprechendsten Algorithmen – z. B. VQE (Variational Quantum Eigensolver) [12], verschiedene QNN-Varianten (Quan-

Abb. 1 Schematische Darstellung des Prinzips der hybriden quantenklassischen Algorithmen. Ein Algorithmus, der auf einem klassischen Rechner ausgeführt wird, steuert die Parametrierung eines Quantenalgorithmus nach einer vorgegebenen Zielfunktion, wobei der Quantenalgorithmus und der klassische Algorithmus abwechselnd ablaufen. (Orientiert an [9])

tum Neural Network) [13] oder auch QAOA (Quantum Approximate Optimization Algorithm) [14], und bedienen mit Simulation, maschinellem Lernen und kombinatorischer Optimierung drei potenzielle Anwendungsgebiete moderner Quantenalgorithmen. Explizit soll aufgrund der hohen Relevanz der Problemstellung noch die Lösung linearer Gleichungssysteme genannt werden. Die Anwendungen sind vielfältig; ein Beispiel wäre die effizientere Implementierung von Finite-Elemente-Verfahren [15]. Der HHL-Algorithmus verspricht hier erhebliche Potenziale, die jedoch aufgrund verschiedener Restriktionen zumindest aktuell nicht genutzt werden können. Abhilfe sollen aber auch für diese Problemstellung hybride quantenklassische Algorithmen schaffen [16].

Dabei ist die Leistungsfähigkeit gegenwärtiger Implementierungen dieser Ansätze auf aktueller Hardware noch nicht in die Nähe klassischer Verfahren gelangt, aber die Liste der verfügbaren Algorithmen ist bereits lang und die Entwicklung schreitet zügig voran. Es besteht also heute – 2020 – Grund zum vorsichtigen Optimismus.

Wie bereits vor 40 Jahren erkannt, wächst der Ressourcenverbrauch zur Simulation von Quantensystemen mit klassischen Rechnern unter Anwendung aktueller (und wohl auch künftiger) Verfahren exponentiell, während Quantenrechner hier potenziell effizient arbeiten, woraus sich nach wie vor Potenziale ableiten. Der Optimismus projiziert sich heute ganz im Einklang mit dem Zeitgeist insbesondere auch auf ML-Algorithmen, die neben verschiedenen Varianten der schon genannten QNN wie Convolutional NN oder Rekurrenten-NN auch Quantenvarianten von SVMs (Support Vector Machines), PCA (Principle Component Analysis - Hauptkomponentenanalyse), Boltzmann-Maschinen, Reinforcement Learning und Bayes'scher Inferenz umfassen. Unter Nutzung quantenspezifischer algorithmischer Strategien wie Amplitudenverstärkung bzw. unter Anwendung einer kompakten Kodierung von Daten in Amplituden weisen sie gegenüber ihren klassischen Verwandten gemäß theoretischer Analysen polynomielle oder gar superpolynomielle Vorteile auf.

Potenziale, Einordnung und antizipierte Entwicklung

Erhebliche Potenziale des Quantencomputing-Ansatzes und der entsprechenden Algorithmen bestehen in eben diesen genannten Vorteilen des Laufzeitverhaltens und der Speicheranforderungen von Algorithmen. Darüber hinaus kommen einige Arbeiten zu dem Ergebnis, dass Quantenverfahren auch andere Vorzüge gegenüber klassischen Algorithmen haben können, beim maschinellen Lernen etwa in puncto Generalisierungsfähigkeit [10].

Einordnung Die Vorteile von Quantenverfahren basieren dabei im Wesentlichen auf der Nutzung der Ressourcen Superposition und Verschränkung als grundlegenden Eigenschaften der Quantenwelt, zu denen es bei den Berechnungsmodellen klassischer Computer keine Äquivalente gibt. Die Umsetzung dieser Idee ist durchaus subtil, wie z.B. das Gottesman-Knill-Theorem mit dem Resultat der effizienten klassischen Simulierbarkeit von ausschließlich auf Clifford-Gates basierten Quantenschaltkreisen zeigt. Gute Quantenalgorithmen nutzen diese Eigenschaften jedoch geschickt aus, um Quantenzustände – also den Speicherzustand des Quantencomputers – so zu manipulieren, dass das Ergebnis der gewünschten Rechnung möglichst zügig erzeugt wird, was bei vielen Verfahren nur mit einer bestimmten Wahrscheinlichkeit gelingt: Quantencomputing ist inhärent probabilistisch. Die Komplexitätsklasse der Probleme, für deren Lösung es einen effizienten Quantenalgorithmus gibt, heißt BQP (Bounded Error Quantum Polynomial Time) – die Klasse der Probleme, die auf einem Quantenrechner mit Fehlerwahrscheinlichkeit höchstens $1/3$ in Polynomialzeit lösbar sind. BQP enthält die klassischen Komplexitätsklassen P, da Quantencomputer klassische Rechner effizient simulieren können; BPP (Bounded Error Probabilistic Polynomial Time) – das „klassische Äquivalent“ zu BQP – jedoch nach allem was wir wissen und annehmen (etwa $P \neq NP$) nicht die Klasse der NP-vollständigen Probleme. Diese bleiben also auch für Quantencomputer „schwierig“ [5].

Weiterhin ist bekannt, dass BQP in der Klasse PSPACE der klassisch mit einem polynomiellen Platzbedarf lösbar Probleme enthalten ist, und der Beweis dessen durch die Simulation eines Quantenschaltkreises mit einem klassischen Rechner macht direkt klar, dass Quantencomputer die Klasse der entscheidbaren Sprachen nicht erweitern, da sie eben allesamt von klassischen Rechnern simuliert werden können; vermutlich aber eben nicht alle auch effizient [17]. Somit sind die Grenzen des Quantencomputing-Ansatzes in Bezug auf Effizienz und Berechenbarkeit in diesem Rahmen abschätzbar. Was bleibt sind signifikante Vorteile für Quantenrechner bei der Anwendung auf Probleme wie der Faktorisierung natürlicher Zahlen, die – vermutlich – nicht in BPP, wohl aber in BQP liegen. Weitere relevante Vertreter dieser Problemklasse gilt es zu identifizieren. Klar ist auch, dass diese Komplexitätsbetrachtungen polynomielle Speed-ups in beliebiger Form erlauben, was praktisch dann erheblichen Vorteilen entsprechen kann.

Für einige Verfahren des maschinellen Lernens werden, wie beschrieben, Speed-ups unterschiedlichen Ausmaßes erwartet. Bei Problemstellungen der Optimierung geht man von ähnlichen Resultaten aus. Komplexitätsbetrachtungen des Quantencomputings sind heute in vielen Details untersucht (vgl. [18] zur Übersicht). Beachten Sie hier auch das spannende und in mehrfacher Hinsicht weitreichende Re-

sultat $MIP^* = RE$ einer neuen Charakterisierung der Menge der rekursiv aufzählbaren Sprachen durch Ji et al. vom Januar 2020 [19]. Es ist neben seiner direkten Relevanz in der Informatik bzw. der Komplexitätstheorie auch in der (Quanten-)Physik – *die Arbeit liefert auch eine vielfach unerwartete negative Antwort auf Tsirelsons Problem im Kontext von Korrelation und Verschränkung* – und der Mathematik – *die Arbeit liefert auch eine Falsifizierung der nach Fields-Medaillist Alain Connes benannten Connes-Vermutung in der Theorie der von Neumann Algebren* – von besonderem Interesse und wurde bereits rege kommentiert.

Antizipierte Entwicklung Einen Blick in die Zukunft wollen wir anhand der Entwicklung bei der Quantenhardware wagen. Dass die Entwicklung von Quantenhardware gegenwärtig große Fortschritte macht, ist unbestritten; dass sie diese auch weiterhin macht, ist eine Wette auf die Zukunft – jedoch erwartbar, wenn die zugrundeliegende Quantentheorie die Welt korrekt beschreibt. Damit ist die praktische Entwicklung immer leistungsfähigerer Quantenrechner übrigens auch ein spannendes physikalisches Experiment. Die Möglichkeit, dabei auf Abweichungen von unseren aktuellen Sichtweisen in der (Quanten-)Physik zu stoßen, finden nicht wenige als Gegenpol zur Aussicht auf die beschriebenen Möglichkeiten eines skalierbaren, leistungsfähigen Quantencomputings besonders spannend – vgl. etwa Scott Aaronson [17] oder Physik-Nobelpreisträger Gerard 't Hooft mit seiner Interpretation der Quantenmechanik auf der Basis Zellulärer Automaten, die – falls korrekt – skalierbares Quantencomputing nicht zulassen würde [20].

Prinzipiell verdoppelt sich – von Verlusten durch Fehlerkorrektur usw. einmal abgesehen – die Leistung eines Quantencomputers mit der Hinzunahme eines einzelnen weiteren Qubits. Eine Größe, die die Leistungsfähigkeit von Quan-

tenrechnern repräsentieren soll und neben der Anzahl der Qubits auch Fehlerraten, die Konnektivität zwischen Qubits sowie darüber hinausgehende Parameter einbezieht, ist das sogenannte Quantenvolumen [21]. Dessen exponentielles Wachstum soll das Moor'sche Gesetz ablösen, das jahrzehntelang die Schlagzahl bei der Weiterentwicklung von Computerhardware bestimmt hat. IBM strebt eine jährliche Verdopplung dieser Leistungskenngröße an – also im Wesentlichen die jährliche Verdopplung der Leistungsfähigkeit von Quantenrechnern. Beim 28-Qubit-Rechner IBM Raleigh liegt sie derzeit bei 32 (Stand Januar 2020) [22]; Honeywell beansprucht aktuell ein Quantenvolumen von 64 für den eigenen Quantenrechner (Stand Juni 2020) und hat sich eine – Achtung! – jährliche Verzehnfachung vorgenommen [23] (vgl. Abb. 2). Man darf gespannt sein, ob diese Zielsetzungen tatsächlich Realität werden und ob sich das Quantenvolumen als Kenngröße zur Leistungsmessung bewährt und bei weiteren Herstellern etabliert.

Ausblick und Zusammenfassung

Schnallen wir uns also an und genießen den Flug! Neben den genannten Protagonisten gibt es eine Reihe weiterer, die ebenfalls an eigenen Quantenmaschinen arbeiten. Microsoft arbeitet beispielsweise am vielversprechenden Ansatz des topologischen Quantencomputings; weitere Protagonisten beim Thema Hardware sind AQT, IonQ, Xanadu, etc. Neben diesen Entwicklungen und Fortschritten bei den Algorithmen wie oben beschrieben, sind auch beim Thema Programmiersprachen Neuigkeiten zu verzeichnen (vgl. Q#, Silq oder schlicht die vielfältige Integration von Quantencomputing-Artefakten in Python), womit die heutige Entwicklung auch an das anknüpft, was bereits im Schlagwort

Abb. 2 Darstellung der Zunahme des Quantenvolumens für die IBM-Q Maschinen und für die Quantenrechner der Firma Honeywell für die jeweils aktuellen Quantenrechner bzw. die Quantenrechner der vergangenen Jahre sowie die durch die beiden Firmen antizipierte Entwicklung des Quantenvolumens gemäß den eigenen Vorhersagen [22, 23]

von Roland Rüdiger im Jahre 2003 thematisiert wurde [24]. Dies treibt in Summe auch die Entwicklung der Softwareframeworks voran (vgl. Cirq, Forest SDK, PennyLane, Qiskit, TensorFlow Quantum) sowie das Angebot von zunehmend plattformübergreifenden Clouddiensten, die Quantencomputing auf der Basis von Simulatoren oder realer Quantenhardware als Service anbieten (z. B. Azure Quantum, amazon Bracket). Quantenrechner werden – aller Voraussicht nach – weiterhin an Leistung gewinnen und damit potenziell immer relevanter. Dabei dürften sie zukünftig Bereiche der Überlegenheit gegenüber klassischen Rechnern nach den ersten Resultaten in dieser Hinsicht zunehmend deutlicher markieren.

Quantencomputing hat sich damit seit der Entstehung der Idee vor 40 Jahren und einer schrittweisen Entwicklung des Gebiets zunächst auf theoretischer Ebene zu einem dynamischen Feld praktischer Forschung mit der Untersuchung anwendungsrelevanter Problemstellungen und der praktischen Implementierung von Algorithmen entwickelt.

Innerhalb der Informatik-GI wurde kürzlich der Arbeitskreis Quantencomputing als Plattform des Austauschs und der Zusammenarbeit zu Themen im Bereich Quantencomputing gegründet. Bitte wenden Sie sich bei Interesse an einer Mitarbeit direkt an Prof. Jörg Lässig unter der E-Mail-Adresse jlaessig@hszg.de.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

Literatur

1. Hertrampf U (2000) Quantencomputer. Informatik Spektrum 5:322–324
2. Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. Proc. 35th Annual Symp. on the Foundations of Computer Science: 124–134. <https://doi.org/10.1109/SFCS.1994.365700>
3. Grover LK (1996) “A fast quantum mechanical algorithm for database search.” Proceedings of the twenty-eighth annual. ACM, symposium on Theory of computing
4. <https://quantumalgorithmzoo.org>. Zugegriffen: 13. Aug. 2020
5. Nielsen MA, Chuang IL (2001) Quantum computation and quantum information. Phys Today 2(54):60
6. Johnson MW et al (2011) Quantum annealing with manufactured spins. Nature 7346:194–198
7. Debnath S et al (2016) Demonstration of a small programmable quantum computer with atomic qubits. Nature 536(7614):63–66
8. Preskill J (2018) Quantum Computing in the NISQ era and beyond Bd. 2, S 79
9. Arute F et al (2019) Quantum supremacy using a programmable superconducting processor. Nature 574(7779):505–510
10. Schuld M, Petruccione F (2018) Learning with Quantum Models. Supervised Learning with Quantum Computers. Springer, Cham, S 247–272
11. McClean JR et al (2016) The theory of variational hybrid quantum-classical algorithms. New J Phys 18(2):23023
12. Parrish RM et al (2019) Quantum computation of electronic transitions using a variational quantum eigensolver. Phys Rev Lett 122(23):230401
13. Farhi E, Neven H (2018) Classification with quantum neural networks on near term processors. Arxiv Prepr Arxiv 1802:6002
14. Farhi E, Goldstone J, Gutmann S (2014) A quantum approximate optimization algorithm. Arxiv Prepr Arxiv 1411:4028
15. Montanaro A, Pallister S (2016) Quantum algorithms and the finite element method. Phys Rev, A 93(3):32324
16. Chen C-C et al (2019) Hybrid classical-quantum linear solver using Noisy Intermediate-Scale Quantum machines. Sci Rep 9(1):1–12
17. Aaronson S (2013) Quantum computing since Democritus. Cambridge University Press, New York
18. <https://complexityzoo.uwaterloo.ca>. Zugegriffen: 13. Aug. 2020
19. Ji Z et al (2020) MIP*= RE. Arxiv Prepr Arxiv 2001:4383
20. ’t Hooft G (2016) The cellular automaton interpretation of quantum mechanics. Springer Nature, Berlin, Heidelberg, New York
21. Bishop, Lev S., et al. “Quantum volume.” Quantum Volume. Technical Report (2017)
22. Moore SK (2020) Honeywell claims it has most powerful quantum computer. IEEE Spectr. <https://spectrum.ieee.org/tech-talk/computing/hardware/honeywell-claims-it-has-most-powerful-quantum-computer>
23. <https://www.ibm.com/blogs/research/2020/01/quantum-volume-32/>. Zugegriffen: 13. Aug. 2020
24. Rudiger R (2003) Quantenprogrammiersprachen. Informatik Spektrum 6:406–409