Primzahlen als Herausforderung

Reinhard Kahle[1]

Abstract: Wir diskutieren, in welchem Maße zahlentheoretischen Fragestellungen von der neuen, statistikbasierten Künstlichen Intelligenz gelöst werden könnten und welche Auswirkungen das z.B. in der Kryptographie hätte. Dabei sehen wir die wesentliche Herausforderung darin, Methoden zu finden, die zu zeigen erlauben, was diese KI *nicht* leisten kann.

Keywords: Zahlentheorie; Neue KI; Diskrete Probleme; RSA

1 Primzahlen

Schon Euklid konnte beweisen, daß es unendlich viele Primzahlen gibt. Doch ist es Mathematikern bis heute nicht gelungen, eine eingängige Struktur in der Primzahlverteilung zu finden, die es über eine rein statistische Vorhersage über die Anzahl der Primzahlen in einem Intervall erlauben würde, natürliche Zahlen „einfach“ auf ihre Primzahleigenschaft hin zu prüfen. Das *Sieb des Erathostenes* liefert zwar eine sichere Methode, um zu überprüfen, ob eine Zahl prim ist, nur ist dieses Verfahren bei hinreichend großen Zahlen hoffnungslos aufwendig; und auch wenn wir heute bessere Methoden kennen, gibt es keine, die nach den in der Komplexitätstheorie entwickelten Kriterien als *schnell* eingestuft werden kann. Das gilt auch nicht für die Methode, die dem bedeutenden Resultat zugrunde liegt, daß die Primzahleigenschaft *polynomiale Berechnungskomplexität* hat [AKS04]; dabei erweist sich nicht nur der Grad des Polynoms für die Laufzeit als problematisch, sondern auch die Platzkomplexität. In der Praxis läßt sich aber zum Glück mit probabilistischen Methoden für Zahlen in der aktuell relevanten Größenordnung der Kryptographie in ausreichender Zeit und mit hinreichender Sicherheit überprüfen, ob diese prim sind (*Miller-Rabin-Test*, [Ra76]) .

Die Bedeutung, die Primzahlen insbesondere in der Kryptographie haben, legt nahe, daß es eine interessante Frage sein sollte, ob es der modernen, auf statistischen Methoden basierenden KI gelingen könnte, einen *schnellen* und zuverlässigen Primzahltest zur Verfügung zu stellen. Dazu gibt es Untersuchungen (siehe z.B. [ES06]), die aber – soweit wir wissen – zur Zeit noch nicht „in Konkurrenz“ zu den etablierten Algorithmen treten können.

[1] Carl Friedrich von Weizsäcker-Zentrum, Universität Tübingen, Keplerstr. 2, 72074 Tübingen, Deutschland, und CMA, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal, kahle@mat.uc.pt

doi:10.18420/inf2020_64

Aus theoretischer Sicht ist die Suche nach einem Primzahltest durch die „neue KI“ eine Aufgabe, die sich in einem größeren Kontext stellt: In welchem Maße kann die KI *Strukturen entdecken*, die Mathematikern (bisher) verborgen geblieben sind.

Es gilt als Stärke der neuen KI, daß sie in der Lage ist, bei geeignetem Training in großen Datenmengen („Big Data“) *Strukturen* zumindest implizit ausfindig zu machen, die es erlauben, anschließend auch Einzeldaten im Hinblick auf verschiedene Eigenschaften hin zu klassifizieren.

2 Teilbarkeit

In unserem Kontext wollen wir derartige Strukturen innerhalb der Arithmetik ausnutzen. Wenn wir hier von Strukturen sprechen, meinen wir Gesetzmäßigkeiten, die sich in mathematischen Objektbereichen, wie z.B. den natürlichen Zahlen, aufweisen lassen.

Die Eigenschaft, gerade zu sein, läßt sich für eine natürlich Zahl bereits dadurch nachprüfen, daß man nachsieht, ob die letzte Ziffer dieser Zahl (in Dezimaldarstellung) 0, 2, 4, 6 oder 8 ist; für die Teilbarkeit durch drei ist es ausreichend, zu prüfen, ob die Quersumme der untersuchten Zahl durch drei teilbar ist. Im Fall der Primzahlen, für die zwar das Sieb des Erathostenes als Algorithmus zur Verfügung steht, ist es aber eine offene Frage, ob es eine in den natürlichen Zahlen verborgene Struktur gibt, die einen „einfachen“ Primzahltest ermöglichen würde.

Ohne daß wir das effektive durchgeführt hätten, gehen wir hier davon aus, daß die statistikbasierte KI in der Lage sein sollte, zu *lernen*, wann eine natürliche Zahl gerade oder durch drei teilbar ist.

Wenn eine KI-Software die Gradzahligkeit tatsächlich so gelernt hat, daß nur die letzte Ziffer getestet wird, sollte sich das im Prinzip durch Laufzeitüberprüfung feststellen lassen – ohne daß dafür die *Black Box* geöffnet werden müßte.

Im Fall der Teilbarkeit durch drei ist nicht klar, ob sich die Form, in der der Test von einer KI-Software durchgeführt wird, rekonstruieren ließe. Aber es auch nicht zu sehen, warum die Software die Idee der Quersumme, wenn auch nur implizit, verwenden sollte.

Allgemein können die Teilbarkeitstests für verschiedene Zahlen sehr unterschiedlich ausfallen, so daß man fragen kann, wie die KI hier eigentlich vorgehen dürfte. Hat sie z.B. eine Möglichkeit, aus den trivialen Teilbarkeitstest für 2 und 5 *selbständig* den trivialen Test für 10 zu erlernen? Kann Sie *bessere* Tests für die Teilbarkeit durch 17 finden, als eine *brute-force* Methode, die man auf Grundlage der Schulmathematik verwenden würde?

In diesem Zusammenhang sei auch kurz auf ein „Problem“ der Mathematik in der Aufstellung

ihrer Hilfsformeln verwiesen: Für die Determinantenberechnung einer gegebenen Matrix mit Hilfe des Laplaceschen Entwicklungssatzes werden gerne Formeln der Form:

$$\det A = \sum_{i=1}^{n} (-1)^{i+j} \cdot a_{ij} \cdot \det A_{ij}$$

benutzt. Der Ausdruck $(-1)^{i+j}$ bestimmt dabei das Vorzeichen eines jeden Summanden. In einer naiven algorithmischen Ausführung würde er $i + j$ viele Multiplikationen nach sich ziehen, die offensichtlich unnötig sind, da man lediglich eine Fallunterscheidung braucht, die prüft, ob $i + j$ gerade oder ungerade ist. Da die übliche Formelsprache der Arithmetik aber keine Fallunterscheidung kennt, bedient man sich eines formal korrekten Ausdrucks, der aber einen „algorithmischen Overkill" darstellt. Wenn wir entsprechende funktionale Zusammenhänge von einer KI-Software lernen lassen, ist zu erwarten, daß diese sich nicht an tradierten Darstellungsformen orientiert, sondern beliebige algorithmische Lösungen finden kann. Wie der benutzte Algorithmus konkret aussieht, würde man erst wissen, wenn man die der KI zugrundeliegende *Black Box* öffnen könnte. Doch könnten Laufzeitbetrachtungen eventuell gewisse Hinweise bringen, auch wenn sich die *Black Box* nicht öffnen läßt.

Dementsprechend geht es uns hier nicht darum, die von der KI implizit verwendeten Strukturen bzw. Algorithmen, explizit zu machen (also die *Black Box* zu öffnen); im Gegenteil, es geht gerade um die Beurteilung der (potentiellen) Leistungsfähigkeit der KI, wenn man ihre detaillierte Vorgehensweise *nicht* kennt.

Was die Primzahlen betrifft, könnte die Hoffnung sein, daß die KI einen *einfachen* Test erlernen könnte, der uns einen *schnellen* Primzahltest zur Verfügung stellen würde. Eine solche Hoffnung wäre mit der Überzeugung verbunden, daß die Primzahleigenschaft auf einer *einfachen* strukturellen Eigenschaft beruht, die die Mathematiker bisher nicht entdeckt haben, die aber die KI entdecken könnte. Allerdings kann es natürlich auch der Fall sein, daß eine solche einfache Eigenschaft gar nicht existiert und alle Versuche daher umsonst wären. Zunächst wollen wir auf ein „Standardargument" eingehen, warum die neue KI nicht spezifisch für Primzahltests ausgelegt sein sollte, und das auf den Unterschied von diskreten und kontinuierlichen Fragestellungen verweist.

3 Diskrete versus kontinuierliche Probleme

Hermann Weyl gab dem für uns wichtigen Unterschied von diskreten und kontinuierlichen Fragestellungen eine sehr schöne Einkleidung, allerdings unter der Annahme, daß die zahlentheoretischen Eigenschaften im wesentlichen nur einer Zahlenmagie dienen – die folgenden Zeilen wurden vor der Entdeckung der Bedeutung der Zahlentheorie für die Kryptographie geschrieben [We71, S. 36f]:

> PLATO übernimmt ein gut Stück der pythagoreischen Zahlenweisheit; aber die Zahl der Bürger einer idealen Stadt, welche er zu 5040 = 7! ansetzt . . . schein[t] seine eigene numerlogische Erfindung zu sein. . . .
>
> Ich möchte nur auf einen Zug hinweisen, der für diese Denkweise kennzeichnend zu sein scheint: was in der Zahlenmagie gilt, sind die *zahlentheoretischen* Eigenschaften der Zahlen; was in der Naturwissenschaft gilt, sind ihre *Größen*eigenschaften. Vom Größen-Standpunkt macht es wenig Unterschied, ob die Zahl der Bürger einer Stadt 5040 beträgt oder 5039, vom zahlentheoretischen Standpunkt ist da ein himmelweiter Abstand; z.B. besitzt $5040 = 2^4 \cdot 3^2 \cdot 5 \cdot 7$ viele Teiler, während 5039 eine Primzahl ist. Wenn in PLATOS idealer Stadt *ein* Bürger über Nacht stirbt und dadurch die Bürgerzahl auf 5039 erniedrigt, ist sie sogleich völlig korrumpiert.

Aber nicht nur Platos ideale Stadt hätte unter einem solchen kleinen Unterschied zu leiden, jedes moderne kryptographische Protokoll würde in gleicher Weise korrumpiert.

Wenn nun die neue KI ganz wesentlich auf kontinuierliche (stetige) Eigenschaften der von ihr (implizit) benutzten Funktionen aufbaut, könnte dies durchaus ein Argument sein, warum zahlentheoretische Fragestellungen dieser KI nicht unmittelbar zugänglich sind.

Allerdings hat auch die Zahlentheorie erkannt, daß sich tiefliegende Erkenntnisse über natürliche Zahlen gewinnen lassen, wenn man reelle Zahlengrößen mit hinzunimmt. Euler begründete die *analytische Zahlentheorie*, in der Methoden der Differential- und Infinitesimalrechnung erfolgreich in der Zahlentheorie zur Anwendung kommen. Insbesondere eröffnete sich damit die Möglichkeit zur Formulierung und zum Beweis des Primzahlsatzes, der besagt, daß sich die Anzahl der Primzahlen unterhalb von x asymptotisch an $\frac{x}{\ln(x)}$ annähert.

Im Kontext der analytischen Zahlentheorie läßt sich zumindest nicht mehr mit dem Unterschied von diskreten und kontinuierlichen Fragestellungen argumentieren. Trotzdem ist es – nach unserem Wissen – eine offene Frage, welche zahlentheoretische Probleme der neuen KI zugänglich sind. Im Hinblick auf bekannte Resultate wäre es durchaus interessant zu sehen, ob sich der Primzahlsatz auch durch reine KI-Analyse gewinnen ließe, ebenso wie viele der überraschenden zahlentheoretischen Zusammenhänge, die vor allem Euler und der geniale indische Mathematiker Ramanujan gefunden haben. Noch spannender wäre es natürlich, wenn die KI zur Lösung von offenen Fragen beitragen könnte, z.B. ob es nur endlich viele oder unendlich viele Primzahlzwillinge gibt.

Unser Interesse hier richtet sich jetzt aber auf die Frage, was sich *nicht* berechnen, oder zumindest *nicht effizient* berechnen läßt. Diese Frage läßt sich im Diskreten sehr detailliert untersuchen.

4 Berechenbarkeits- und Komplexitätstheorie

Die formale Berechenbarkeitstheorie für Funktionen auf den natürlichen Zahlen wurde vor allem im Anschluß an Hilberts *Entscheidungsproblem* sowie das für die Klärung der Grundlagen der Mathematik entwickelte *Hilbertsche Programm* ausgearbeitet. Dabei hat sich das von Turing aufgestellte Berechenbarkeitsmodell der *Turing-Maschine* besonders bewährt. Die Beobachtung, daß alle alternativen Berechenbarkeitsmodelle äquivalent zu Turings Modell sind (soweit sie sich nicht als schwächer herausgestellt haben), hat ihren Niederschlag in der *Churchen These* gefunden, daß der (informale) Berechenbarkeitsbegriff durch die Turing-Maschinen (oder jedes äquivalente Modell) bereits erschöpfend charakterisiert ist.

Damit haben wir ein Werkzeug an der Hand, mit dem die Grenzen der Berechenbarkeit formal erfaßt werden können. Insbesondere ließ sich mit Hilfe eines Diagonalisierungsarguments zeigen, daß das *Halteproblem*, d.h. die Frage, ob ein gegebenes Programm zur Berechnung einer partiellen Funktion auf den natürlichen Zahlen für einen gegebenen Eingabewert ein Ergebnis liefert oder nicht, unentscheidbar ist.

Bemerkung Die in der Churchen These zum Ausdruck kommende Grenzziehung war den an der Entwicklung der Berechenbarkeitstheorie beteiligten Protagonisten nicht von vorneherein klar. So hatte Gödel im Rahmen des Beweises seiner Unvollständigkeitssätze den (Turing-vollständigen) Berechenbarkeitsbegriff der *rekursiven Funktionen* herausgearbeitet. Über die Tragweite der Unvollständigkeitssätze war er sich aber bei der Abfassung seiner epochalen Arbeit [Gö31, S. 197] noch nicht im klaren, als er schrieb:

> Es sei ausdrücklich bemerkt, daß Satz XI [*der zweite Gödelsche Unvollständigkeitssatz*] ... in keinem Widerspruch zum Hilbertschen formalistischen Standpunkt steh[t]. Denn dieser setzt nur die Existenz eines mit finiten Mitteln geführten Widerspruchsfreiheitsbeweises voraus und es wäre denkbar, daß es finite Beweis gibt, die sich in *P* [die von Gödel benutzte Axiomatisierung der Arithmetik] *nicht* darstellen lassen.

In einer Bemerkung, die Gödel 1963 einer englischen Übersetzung seines Artikels hinzufügte, gesteht er dann zu, daß es (erst) Turings Analyse des Berechenbarkeitsbegriff war, die ihn von der uneingeschränkten Gültigkeit seines Resultates überzeugte [Gö67, S. 616]:

> In consequence of later advances, in particular of the fact that due to A. M. Turing's work a precise and unquestionably adequate definition of the general notion of formal system can now be given, a completely general version of Theorems VI and XI is now possible.

Wir haben diese historische Bemerkung hier aufgenommen, um zu zeigen, daß sich die Entwicklung einer neuen Theorie, wie wir sie hier als Herausforderung für die KI formulieren,

sicherlich nicht gradlinig und direkt erfolgen wird, sondern erst durch ein Wechselspiel von unterschiedlichen Ansätzen gewonnen werden kann. ⊣

Über ihre Funktion für Unentscheidbarkeitsresultate hinaus hat sich das Konzept der Turing-Maschine vor allem auch dadurch bewährt, daß es eine eingängige Definition von Berechenbarkeitskomplexität ermöglichen. Für einen gegebenen Eingabewert kann man die Schritte zählen, die eine gegebene Turing-Maschine benötigt, um ein Ergebnis zu liefern. Die Komplexität für den durch die Turing-Maschine bestimmten Algorithmus zur Berechnung einer zahlentheoretischen Funktion ergibt sich dann, salopp formuliert, durch eine Funktion, die die entsprechende Schrittzahl majorisiert. Damit kann man z.B. die Klasse *Ptime* der *polynomialen* Funktionen definieren, die Algorithmen besitzen, deren Schrittzahl sich polynomial beschränken läßt. Bekannt ist auch die Klasse *NP*, bei der die Lösung eines mit n parametrisierten Problems *geraten* werden kann, die Überprüfung der Richtigkeit der Lösung dann aber nur polynomiale Laufzeit in n benötigen darf. Wenn mit P die Klasse der polynomial berechenbaren Entscheidungsproblemen bezeichnet wird, ist die Frage $P \stackrel{?}{=} NP$ das zentrale offene Problem der theoretischen Informatik. Für die Verläßlichkeit der im folgenden diskutierten kryptographischen Protokolle geht man aber von der allgemein vermuteten Ungleichheit $P \neq NP$ aus. Ungeachtet unserer Bemerkung zum polynomialen Primzahltest werden in diesem Fall Probleme und Funktionen in P beziehungsweise *Ptime* als (im Prinzip) effizient lösbar beziehungsweise berechenbar betrachtet, während Probleme in *NP* zu den (mit heutigen Mitteln) nicht effizient lösbaren Problemen gehören.

Wir haben diese wohl allgemein bekannten, grundlegenden Konzepte der Komplexitätstheorie hier ausgeführt, weil wir im abschließenden Kapitel die Frage aufwerfen werden, was die entsprechende Komplexitätstheorie für die KI sein könnte, beziehungsweise, was das Fehlen einer solchen Komplexitätstheorie für die KI bedeutet.

5 Sind unsere kryptographischen Protokolle vor der neuen KI sicher?

Die modernen *public-key*-Protokolle der Kryptographie basieren zentral auf zahlentheoretischen *Einbahnstraßenfunktionen*: $f(x) = y$ soll einfach berechenbar sein (polynomial); die inverse Funktion $f^{-1}(y) = x$ soll mit öffentlich zugänglicher Information nur schwer (exponentiell) berechenbar sein; und allein mit Hilfe eines privaten Schlüssel ist auch sie „durch eine Hintertür“ einfach berechenbar.

Im bekannten RSA-Protokoll nutzt man an zentraler Stelle aus, daß die Multiplikation zweier Zahlen *einfach* ist, die Zerlegung einer Zahl in seine Primfaktoren aber *schwer*. Mit Hilfe einer schnellen Faktorisierung großer Zahlen würde man dieses Protokoll sofort kompromittieren.

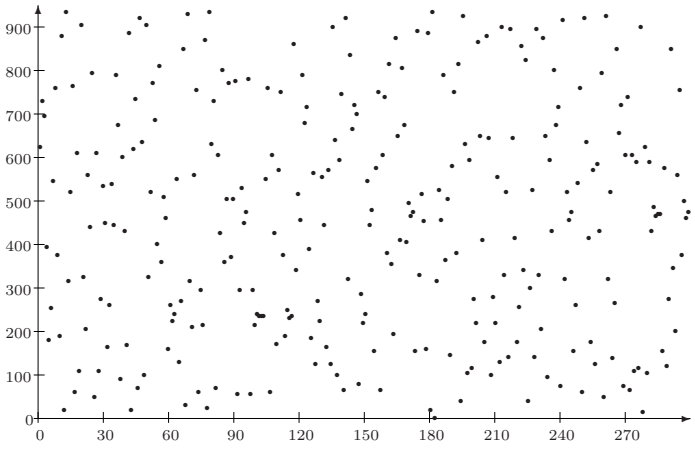
Damit stellt sich die Frage, ob die moderne KI unter Umständen eine solche schnelle Faktorisierung liefern könnte. Nach dem aktuellen Kenntnisstand ist das nicht der Fall und „es sieht auch nicht so aus“, daß die KI hierfür einen besonderen Ansatzpunkt hätte.

Aber haben wir eine Möglichkeit, den *Eindruck*, daß RSA – unter der vorausgesetzten Annahme von $P \neq NP$ – von Seiten der KI keine Gefahr drohe, durch formale Argumente zu untermauern? Anders gefragt: Was wäre die adäquate Berechenbarkeitstheorie für die neue KI, die es erlaube, formale Unentscheidbarkeits- und Komplexitätsresultate zu zeigen, wie wir sie aus der Turing-Berechenbarkeit kennen?

Hierbei sollte eine solche *KI-Berechenbarkeitstheorie* nicht von einer „Öffnung" der *Black Box* abhängen (denn dann sollte sich die traditionelle Berechenbarkeitstheorie auf den „verstandenen" Algorithmus anwenden lassen), sondern sie sollte Ergebnisse liefern, selbst wenn man den internen Ablauf der Entscheidungsfindung nicht kennt.

Eine konkrete Aufgabenstellung läßt sich am *Diskreten Logarithmus-Problem* (DLP) illustrieren, das dem *Diffie-Hellman-Schlüsselaustausch* und *ElGamal-Protokoll* zugrunde liegt. In einem endlichen Körper der Charakteristik p (p prim), läßt sich für eine gegebene primitive Wurzel g und gegebene Zahl x der Wert h in $g^x \equiv h \pmod{p}$ *schnell* berechnen. Umgekehrt ist die Berechnung von x bei gegebenen g und h aber – nach aktuellem Kenntnisstand – bei ausreichend großem p nicht mehr praktisch durchführbar.

In der folgenden Tabelle (entnommen aus [HPS08, S. 65]) sind die Werte von $627^i \bmod 941$ für $i = 1, 2, 3, \ldots$ eingetragen:



Während sich 627^i durch eine „übersichtliche" Exponentialkurve darstellen läßt, führt das „Herunterbrechen" der Funktionswerte an den ganzzahligen Punkten durch die mod 941-Operation zu einem Bild, in dem sich unmittelbar keine einfache Struktur mehr erkennen läßt. Würde eine KI-Software hier (und für anderen Zahlen g und p) aber doch eine einfache Struktur erlernen können, hätte sie auch eine Handhabe, um DLP zu lösen (und das unabhängig davon, ob diese Struktur uns zugänglich gemacht werden kann oder nicht).

Wie im Fall der Primzahlen ist es offen, ob eine solche Struktur existiert; aber um sich auf die Sicherheit der DLP-basierten kryptographischen Protokolle verlassen zu können, wäre es wünschenswert, daß man eine formale Analyse dessen hätte, was die KI in diesen (und ähnlichen) Fällen *nicht* leisten kann.

Schließlich wollen wir noch auf einen Problemkreis aufmerksam machen, der sich selbst der klassischen Komplexitätstheorie entzieht. Aus einem Resultat von Pollard [Po74] ergibt sich, daß RSA dann unsicher ist, wenn das dabei verwendete Produkt zweier großer Primzahlen eine Primzahl p benutzt, deren „Vorgänger" $p - 1$ ein Produkt viele kleiner Primzahlen ist. In einem Lehrbuch zur Kryptographie schreiben die Autoren dazu [HPS08, S. 136]:

> From a cryptographic perspective, the importance of Pollard's method lies in the following lesson. Most people would not expect, at first glance, that factorization properties of $p-1$ and $q-1$ have anything to do with the difficulty of factoring pq. The moral is that even if we build a cryptosystem based on a seemingly hard problem such as integer factorization, we must be wary of special cases of the problem that, for subtle and nonobvious reasons, are easier to solve than the general case.

Im Allgemeinen läßt sich nicht ausschließen, daß eine intelligente KI-Software eine „ELIZA $p + 2$"-Methode findet, bei der RSA dann kompromittiert ist, wenn $p + 2$ spezielle Eigenschaften aufweist.

Zusammenfassend argumentieren wir hier nicht dafür, daß die KI neue Wege finden soll, zahlentheoretische Fragestellungen effizient zu lösen, sondern, ganz im Gegenteil, daß die Herausforderung darin besteht, Methoden zu finden, die zu zeigen erlauben, was die KI *nicht* leisten kann. Das könnte zumindest für diejenigen von Interesse sein, die auch weiterhin Internet-Banking ohne Beunruhigung benutzen wollen.

Literatur

[AKS04] Agrawal, M.; Kayal, N.; Saxena, N.: PRIMES Is in P. Annals of Mathematics 160/2, S. 781–793, 2004.

[ES06] Egri, L.; Shultz, T. R.: A Compositional Neural-network Solution to Prime-number Testing. Proceedings of the Annual Meeting of the Cognitive Science 28, 2006.

[Gö31] Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. Monatshefte für Mathematik und Physik 38, S. 173–198, 1931.

[Gö67] Gödel, K.: On formally undecidable propositions of *Principia Mathematica* and related systems I. In (van Heijenoort, J., Hrsg.): From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931. Harvard University Press, S. 596–616, 1967.

[HPS08] Hoffstein, J.; Pipher, J.; Silverman, J. H.: An Introduction to Mathematical Cryptography. Springer, 2008.

[Po74] Pollard, J. M.: Theorems on factorization and primality testing. Proceedings of the Cambridge Philosophical Society 76/3, S. 521–528, 1974.

[Ra76] Rabin, M. O.: Probabilistic algorithms. In (Traub, J. F., Hrsg.): Algorithms and complexity. Academic Press, S. 21–39, 1976.

[We71] Weyl, H.: Über den Symbolismus der Mathematik und mathematischen Physik. In (Reidemeister, K., Hrsg.): Hilbert. Springer, S. 20–38, 1971.

Diese Arbeit wurde u.a. von der Udo Keller-Stiftung und, in eine frühen Phase, von der VolkswagenStiftung im Rahmen des Projekts *Can Software be 'responsible'*? sowie durch die Portugiesische Forschungsgemeinschaft FCT über das *Centro de Matemática e Aplicações*, UID/MAT/00297/2020, gefördert.