Prof. Dr. Ulrich Görtz
Wintersemester 2023/24

Seminar über
Quantencomputer und Kryptographie

1. Organisatorisches

— Siehe auch die Moodle-Seite zum Seminar. —

Vorbesprechung. Donnerstag, 21.9., 14:15 Uhr, im Mathematik-Gebäude (voraussichtlich in S-U-3.01). Weitere Informationen über die Moodle-Seite. Die Verteilung der Vorträge erfolgt dann kurz nach der Vorbesprechung. Die ersten drei Vorträge würde ich auch schon vorher vergeben, damit genügend Zeit für die Vorbereitung ist.

Termin. Dienstags, 14-16 Uhr, N-U-3.05. Achtung: Am 12.12. findet keine Sitzung statt.

ECTS-Punkte. Das Seminar wird als Master-Seminar im Studiengang Lehramt GyGe/BK abgehalten. Für einen erfolgreichen Vortrag erhalten Sie 4 Credit-Punkte. Siehe auch Modulhandbuch Master, S. 23ff.[1].

Kontakt.
ulrich.goertz@uni-due.de.

Die Seminarvorträge sollen an der Tafel gehalten werden und nicht länger als 80 Minuten dauern. Rechnen Sie bei der Planung mit Fragen der Zuhörer:innen und der Organisatoren. Danach stehen circa 10 Minuten für Fragen und eine Rückmeldung zum Vortrag zur Verfügung.

Im Nachgang eine schriftliche Ausarbeitung Ihres Vortrags abzugeben, ist nicht erforderlich (und kann auch nicht als Ersatz für einen erfolgreichen Vortrag dienen).

Für das Seminar gilt (moralische) Anwesenheitspflicht; es wird eine *aktive Teilnahme* erwartet. Für den Fall, dass Sie an einem Termin aus wichtigen Gründen verhindert sind, entschuldigen Sie sich bitte vorher bei der/dem Vortragenden und bei mir.

Erforderliche Vorkenntnisse. Gute Kenntnisse der Linearen Algebra, insbesondere:

- Gruppen, Untergruppen, [Gö1] Abschnitt 8.1[2],
- Rechnen mit Restklassen ganzer Zahlen, [Gö1] Abschnitt 4.2.1[3]
- Vektorräume,
- unitäre Vektorräume, [Gö2], Abschnitt 19.4[4].

Aus der Analysis benötigen wir insbesondere

[1] https://www.uni-due.de/imperia/md/content/didmath/lehre/mhb_ma_gyge_220705.pdf
[2] https://math.ug/la1-ws2021/sect0031.html
[3] https://math.ug/la1-ws2021/sect0018.html
[4] https://math.ug/la2-ss21/sec-blf-slf.html

- die komplexen Zahlen,
- die komplexe Exponentialfunktion, Sinus, Kosinus,
- den natürlichen Logarithmus $\ln(x)$.

Siehe zum Beispiel [Fo].

2. Anforderungen / Wie halte ich einen guten Seminarvortrag?

- Richten Sie die Vorbereitung auf Ihren Vortrag nach dem Grundsatz aus, dass Ihre Zuhörer möglichst viel dabei lernen und „aus dem Vortrag mitnehmen“ sollen.
- Ihr erstes und wichtigstes Ziel sollte sein, die im Vortrag zu behandelnde Mathematik gründlich zu verstehen. Das wird in der Regel eine ganz Menge Zeit in Anspruch nehmen; beginnen Sie daher frühzeitig mit Vorbereitung!
- Stellen Sie viele Fragen (sich und den anderen Seminarteilnehmer:innen). Seien Sie diszipliniert darin, sich zu fragen, warum (ob) die im zugrundeliegenden Text aufgestellten Behauptungen richtig sind. Was lässt sich vereinfachen? Wenn Sie auf Dinge stoßen, die Sie nicht verstehen, fragen Sie die anderen Seminarteilnehmer:innen (diejenigen, die die Vorträge direkt vor oder nach Ihnen halten, haben sich vielleicht schon genau dieselbe Frage gestellt). Vielleicht ist im Seminarprogramm (oder im Buch, Skript, ...) ein Fehler? Wenn Sie damit nicht weiterkommen, stellen Sie die Frage im Moodle-Forum.
- Wenn Sie die Mathematik verstanden haben, sollten Sie bewusst darüber nachdenken, wie Sie die Inhalte in Ihrem Vortrag darstellen möchten. Oft bietet es sich an, den Aufbau zu ändern, Sachen umzustrukturieren, zusätzliche Beispiele einzubauen, die Notation anzupassen, usw. — schließlich ist ein Vortrag etwas ganz anderes als ein geschriebener Text. Diese Restrukturierung ist Teil der Anforderung an Ihren Vortrag. Es genügt nicht, den Text aus Ihrer Quelle an die Tafel zu schreiben.
- Überlegen Sie im Vorfeld, was die Hauptpunkte Ihres Vortrags sind, die jede:r Teilnehmer:in lernen sollte, und berücksichtigen Sie das entsprechend. Sie können nötigenfalls Sachen im Vortrag auslassen (zum Beispiel eine technische Rechnung, die zum Verständnis nicht notwendig ist (schauen Sie sie sich trotzdem genau an, um gegebenenfalls Fragen beantworten zu können!)). Denken Sie bei der Vortragsplanung darüber nach, was sich zum Weglassen eignet, damit Sie nicht am Ende wegen Zeitmangels gezwungen sind, die interessantesten Teile zu überspringen.
- Denken Sie insbesondere darüber nach, was ein guter Einstieg in den Vortrag ist. Wie fügt er sich ins Seminarprogramm ein? Warum sollte man sich für dieses Thema interessieren? Bereiten Sie auch einen guten Abschluss des Vortrags vor.
- Die Sachen, die Sie in der Vorbereitung besonders viel Zeit gekostet haben, sollten Sie nicht auslassen, denn die anderen Teilnehmer:innen haben vermutlich ähnliche Schwierigkeiten beim Verständnis und können gerade an diesen Stellen viel lernen. Wenn doch einmal noch Fragen bei Ihnen offengeblieben sind, gehen Sie damit ehrlich um und versuchen Sie nicht, Probleme beim

Verständnis zu verschleiern.

- Wenn im Text auf eine Übungsaufgabe verwiesen wird, gehört es zur Vorbereitung des Vortrags, diese Aufgabe zu lösen.
- **Aktive Einbindung der Teilnehmer:innen.** Das Seminar lebt von der aktiven Beteiligung aller Teilnehmer:innen an *allen* Vorträgen, nicht nur am eigenen. Die Beteiligung an den anderen Vorträgen ist eine Möglichkeit, Ihre Note im Seminar gegebenenfalls noch zu verbessern.

  Die Beteiligung aller sicherzustellen, ist sowohl eine Aufgabe der Zuhörer:innen als auch der Sprecher:in. Konkret ist daher eine der Anforderungen an einen erfolgreichen Vortrag: Überlegen Sie sich mindestens drei Fragen, die Sie während Ihres Vortrags an das Publikum richten. Es kann sich um offene Fragen oder um Multiple-Choice-Fragen handeln. Wählen Sie Fragen aus, die sich von denjenigen, die dem Vortrag gefolgt sind, relativ schnell beantworten lassen, und die Ihnen ermöglichen, auf einen besonders wichtigen Punkt noch einmal hinzuweisen und/oder helfen zu entscheiden, was Sie vielleicht noch einmal anders/erneut erklären sollten oder worauf Sie im weiteren Verlauf den Fokus legen.

Wenn Sie Ihren Vortrag in den Grundzügen vorbereitet haben, und *spätestens zwei Wochen vor dem Termin Ihres Vortrags* sprechen Sie Ihren Vortrag mit mir durch. Dabei können Sie offene Fragen klären (allerdings möchte ich als den üblichen Weg für Fragen das Forum auf der Moodle-Seite etablieren). Wenn Sie gar keine Fragen haben, sollten Sie noch einmal sehr selbstkritisch hinterfragen, ob Sie die Inhalte des Vortrag mathematisch wirklich hinreichend durchdrungen haben. Es ist für den Verlauf des Vortrag besser und auch für Sie angenehmer, wenn Sie nicht erst bei Nachfragen während des Vortrags feststellen, dass es doch noch Verständnislücken gibt.

Auch wenn es viel Arbeit sein wird, kann und soll es Spaß machen, an dem Seminar teilzunehmen! Dabei steht im Vordergrund, dass alle Teilnehmer:innen gemeinsam etwas lernen. Daher sollten Sie als Sprecher:in Fragen im Vortrag immer willkommen heißen und als Unterstützung dabei sehen, Ihren Vortrag auf die Bedürfnisse der Teilnehmer:innen auszurichten. Umgekehrt: Geben Sie sich als Zuhörer:in Mühe, den Vorträgen der anderen aktiv zu folgen und stellen Sie dort Fragen, wo Sie etwas nicht verstehen.

Nutzen Sie die Möglichkeit, Ihre eigenen Schwerpunkte zu setzen und überlegen Sie sich, wie Sie Ihren eigenen Vortrag so gestalten können, dass Sie damit zufrieden sind.

3. Literatur

Wir richten uns nicht nach einer einzigen Quelle, und für viele Vorträge ist das Programm so ausführlich, dass Sie vermutlich im wesentlichen damit schon auskommen können. Weitere Referenzen sind in den einzelnen Vortragsbeschreibungen angegeben.

Das Buch [Sch] von Scherer deckt den allergrößten Teil des Seminarprogramms ab (und noch einiges mehr), ist aber nicht an allen Stellen mathematisch sorgfältig formuliert. Das Buch [KSV] von Kitaev, Shen und Vyalyi ist eine gute und umfangreiche

Einführung in das Thema "Quantencomputer aus mathematischer Sicht" (allerdings auf Englisch); der Algorithmus von Shor wird dort in Kapitel 13 (Abschnitt 13.2 und folgende) besprochen.

Ein Klassiker ist die Arbeit [Ma] von Manin, die allerdings recht knapp und in Teilen auf einem mathematisch anspruchsvolleren Niveau geschrieben ist.

Die Bachelor-Arbeit von Aktas [Ak] an der Universität Ulm ist ebenfalls einen Blick wert.

Andere Quellen finden Sie im Literaturverzeichnis am Ende.

4. Vorträge

Vortrag 1 (Einführung)**.** Einführung in das Seminarthema.

Vortrag 2 (Gruppen, der Satz von Euler)**.** Wiederholen Sie aus der Linearen Algebra kurz die Begriffe der *Gruppe*, und der *Untergruppe* (siehe zum Beispiel [Gö1] Abschnitt 8.1[5]). Im Rest des Vortrags betrachten wir nur endliche Gruppen.

Definieren Sie, was man unter der Ordnung einer Gruppe und unter der Ordnung eines Gruppenelements versteht. Definieren Sie die Begriffe der Nebenklasse und des Index $[G : H]$ einer Untergruppe H in einer Gruppe G und beweisen Sie den Satz von Euler-Lagrange: [Gö1] Abschnitt 8.5.1[6]

Satz. Sei G eine endliche Gruppe und sei H eine Untergruppe von G. Dann gilt

$$\#G = [G : H] \cdot \#H.$$

Folgern Sie daraus:

Satz. Seien G eine endliche Gruppe und $g \in G$. Dann ist die Ordnung $\mathrm{ord}(g)$ des Elements g ein Teiler der Gruppenordnung $\#G$.

Besprechen Sie als Beispiele die additiven Gruppen $\mathbb{Z}/n\mathbb{Z}$ der Restklassen ganzer Zahlen modulo n, und die multiplikativen Gruppen $(\mathbb{Z}/n\mathbb{Z})^\times$ der Restklassen modulo n, die ein multiplikatives Inverses besitzen. Zeigen Sie insbesondere, dass die letzte Bedingung dazu äquivalent ist, dass es sich um die Restklasse einer zu n teilerfremden Zahl handelt. Siehe zum Beispiel [Gö1] Abschnitt 4.2.1[7]

Definieren Sie die Eulersche φ-Funktion: Für $n \geq 1$ sei

$$\varphi(n) = \#(\mathbb{Z}/n\mathbb{Z})^\times.$$

Mit anderen Worten ist $\varphi(n)$ die Anzahl der zu n teilerfremden natürlichen Zahlen zwischen 1 und n.

Folgern Sie schließlich den

Satz von Euler. Seien $n \in \mathbb{N}_{>0}$ und $a \in \mathbb{Z}$ zu n teilerfremd. Dann gilt

$$a^{\varphi(n)} \equiv 1 \mod n.$$

[5] https://math.ug/la1-ws2021/sect0031.html
[6] https://math.ug/la1-ws2021/sect0034.html#subsec:elem-zt
[7] https://math.ug/la1-ws2021/sect0018.html

(Den Spezialfall dieses Satzes, dass n eine Primzahl ist, nennt man den *Kleinen Satz von Fermat*. Was ist in diesem Fall $\varphi(n)$?)

Vortrag 3 (Das RSA-Verfahren)**.** Als Vorbereitung benötigen wir den folgenden Satz über die "Multiplikativität der Eulerschen φ-Funktion". Die φ-Funktion wurde im vorherigen Vortrag eingeführt. Beachten Sie, dass im folgenden Satz die Voraussetzung, dass m und n teilerfremd sind, essentiell ist.

Satz. Sind $m, n \in \mathbb{N}_{>0}$ zueinander teilerfremd, dann gilt $\varphi(mn) = \varphi(m)\varphi(n)$.

Beweisen Sie den Satz in den folgenden beiden Schritten: (1) Im ersten Schritt beweisen wir eine Version des Chinesischen Restsatzes. Wir betrachten die Abbildung $\pi\colon \mathbb{Z}/mn\mathbb{Z} \to \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$, die eine Restklasse $a \in \mathbb{Z}/mn\mathbb{Z}$ auf das Paar (a, a) abbildet (wobei das linke a als Restklasse in $\mathbb{Z}/m\mathbb{Z}$, das rechte als Restklasse in $\mathbb{Z}/n\mathbb{Z}$ zu verstehen ist). Dies ist eine wohldefinierte Abbildung(!). Sie ist injektiv und daher (als Abbildung zwischen endlichen Mengen mit gleich vielen Elementen) sogar bijektiv. (2) Für $a \in \mathbb{Z}/mn\mathbb{Z}$ gilt $a \in (\mathbb{Z}/mn\mathbb{Z})^\times$ genau dann, wenn $\pi(a) \in (\mathbb{Z}/m\mathbb{Z})^\times \times (\mathbb{Z}/n\mathbb{Z})^\times$ liegt. Es folgt, dass π eine Bijektion zwischen $(\mathbb{Z}/mn\mathbb{Z})^\times$ und $(\mathbb{Z}/m\mathbb{Z})^\times \times (\mathbb{Z}/n\mathbb{Z})^\times$ induziert; insbesondere haben diese Mengen gleich viele Elemente. Siehe [Gö1] Satz 8.57[8].

Erklären Sie, was der Wert von φ auf einer Primzahlpotenz ist. Damit und mit dem Satz lässt sich die φ-Funktion leicht für alle natürlichen Zahlen berechnen, deren Primfaktorzerlegung man kennt.

Im zweiten Teil des Vortrags kommen wir zum RSA-Verfahren.

Erklären Sie kurz die Grundidee der Public-Key-Kryptographie (Kommunikation über einen nicht-abhörsicheren Kanal ohne vorherigen Schlüsselaustausch).

Besprechen Sie ausführlich das RSA-Verfahren, siehe [Bu] Abschnitt 8.3, insbesondere 8.3.1–8.3.3, [Gö1] Bemerkung 8.58[9], Wikipedia[10].

- Wie wird das Verfahren durchgeführt, d.h. welche Schritte werden zur Verschlüsselung und Entschlüsselung durchgeführt?
- Was ist die mathematische Begründung dafür, dass man nach Verschlüsselung und anschließender Entschlüsselung wieder die ursprüngliche Nachricht erhält?
- Was sind die mathematischen Annahmen, auf denen die Sicherheit das Verfahrens beruht?
- Wenn Zeit bleibt, könnten Sie darauf eingehen, dass man das vorgestellt Verfahren in der Praxis noch modifizieren muss, weil es sonst angreifbar ist, siehe die hinteren Abschnitte in [Bu] 8.3, und/oder Wikipedia[11] (3.4 Angriffe gegen das unmodifizierte RSA-Verfahren ("Textbook-RSA")).

[8] `https://math.ug/la1-ws2021/sect0034.html`
[9] `https://math.ug/la2-ss21/sec-integritaetsringe.html#bem:rsa-2`
[10] `https://de.wikipedia.org/wiki/RSA-Kryptosystem`
[11] `https://de.wikipedia.org/wiki/RSA-Kryptosystem`

Vortrag 4 (Der Primzahltest von Miller und Rabin)**.** Um das RSA-Verfahren durchzuführen, müssen wir "große" Primzahlen finden können.

In diesem Vortrag besprechen wir einen Ansatz dafür, und zwar eine Methode, die praxistauglich ist, d.h., die mit kleinen Modifizierungen auch in der Praxis verwendet wird. Man wählt einfach zufällig eine Zahl der richtigen Größe (also beispielsweise mit der richtigen Stellenzahl) und führt dann einen Primzahltest durch, um zu testen, ob es sich tatsächlich um eine Primzahl handelt (oder mindestens mit einer so hohen Wahrscheinlichkeit, dass man bereit ist, damit zu arbeiten).

Der Kleine Satz von Fermat liefert den folgenden simplen Primzahltest: Sei $n \in \mathbb{N}$ eine natürliche Zahl, von der wir feststellen wollen, ob n eine Primzahl ist. Wir wählen zufällig a zwischen 2 und $n-1$. Haben a und n größten gemeinsamen Teiler > 1, dann ist n offenbar keine Primzahl. Sonst sind a und n teilerfremd und wir berechnen a^{n-1} modulo n. Ist das Ergebnis $\neq 1$, dann ist n keine Primzahl. Sonst kann keine sichere Aussage getroffen werden (und man könnte annehmen, dass n eine Primzahl sein könnte). Gegebenenfalls kann man auch den Test mit anderen Zahlen a wiederholen. Es gibt allerdings Zahlen n, die sogenannten Carmichael-Zahlen, die die Eigenschaft haben, dass für alle zu n teilerfremden a gilt, dass $a^{n-1} \equiv 1 \mod n$ ist, obwohl n keine Primzahl ist.

Der Miller-Rabin-Test ist eine Verfeinerung dieser Grundidee und basiert auf dem folgenden Satz.

Satz. Sei $p \in \mathbb{N}$ eine Primzahl. Wir schreiben $p - 1 = 2^s d$ mit ungeradem $d \geq 1$. Sei $a \in \mathbb{N}$ zu p teilerfremd. Dann ist $a^d \equiv 1 \mod p$, oder wenigstens eine der Zahlen

$$a^d, \quad a^{2d}, \quad a^{4d}, \quad \ldots, \quad a^{2^{s-1}d}$$

kongruent zu -1 modulo p.

Beweis. Benutzen Sie den Kleinen Satz von Fermat. □

Aus dem Satz folgt, anders ausgedrückt:

Korollar. Sei $n \in \mathbb{N}$, $n - 1 = 2^s d$ mit ungeradem d, und $a \in \mathbb{N}$ teilerfremd zu p. Ist $a^d \not\equiv 1 \mod n$ und ist keine der Zahlen

$$a^d, \quad a^{2d}, \quad a^{4d}, \quad \ldots, \quad a^{2^{s-1}d}$$

kongruent zu -1 modulo n, dann ist n keine Primzahl.

Das Ergebnis des Miller-Rabin-Primzahltests für eine natürliche Zahl n ist entweder, dass die Zahl n keine Primzahl ist, oder dass die Zahl n mit einer "hohen" Wahrscheinlichkeit eine Primzahl ist. Dabei funktioniert der Test folgendermaßen: Sei $n \in \mathbb{N}$ gegeben. Wir wollen feststellen, ob n eine Primzahl ist. Wir wählen zufällig eine natürliche Zahl zwischen 2 und $n-1$. Zuerst berechnen wir den größten gemeinsamen Teiler von a und n. Ist dieser größer als 1, dann ist n keine Primzahl. Sind andererseits a und n teilerfremd, dann überprüfen wir das Kriterium aus dem vorherigen Korollar. Ist das Kriterium erfüllt, dann ist n mit Sicherheit keine Primzahl. Ist das Kriterium nicht erfüllt, dann kann man keine sichere Aussage treffen. Je nachdem, welche Genauigkeit des Ergebnisses man anstrebt, kann man den Test (für dasselbe n) mit einer anderen Zahl a wiederholen. Wenn man mehrere Wahlen für a

ausprobiert und in keinem Fall das Ergebnis erhält, dass n keine Primzahl ist, dann ist n mit hoher Wahrscheinlichkeit eine Primzahl. Man kann zeigen, dass für jede Nicht-Primzahl n ein zu n teilerfremdes a existiert, so dass das Kriterium aus dem Korollar erfüllt ist; es gibt also – anders als bei dem zu Beginn geschilderten Test – keine Zahlen, für die die Keine-Primzahl-Eigenschaft aus Prinzip nicht festgestellt werden kann.

Erklären Sie die Aussage und den Beweis von Theorem 7.6 in [Bu] zur Abschätzung der Erfolgswahrscheinlichkeit des Miller-Rabin-Tests. Dies ist der mathematisch schwierigste Teil des Vortrags. Dort finden Sie auch noch weitere Details zum Miller-Rabin-Test.

Anmerkungen zu diesem Beweis: Wir betrachten eine ungerade Zahl $n \geq 3$, die keine Primzahl ist und wollen Theorem 7.6 in [Bu] für die gewählte Zahl n beweisen.

Wir definieren d und s durch die Gleichung

$$n - 1 = 2^s d$$

und die Bedingung, dass d ungerade sei. Weil n ungerade ist, ist $n-1$ gerade und folglich $s \geq 1$.

Wie in [Bu] begründet, können wir annehmen, dass $a \in (\mathbb{Z}/n\mathbb{Z})^\times$ und ein $r \in \{0, \ldots, s-1\}$ mit $a^{2^r d} \equiv -1 \mod n$ existieren. Wir definieren k als das größte r (zwischen 0 und $s-1$), für das es ein solches Element a gibt, und schreiben $m = 2^k d$.

Wir definieren eine Kette von Untergruppen

$$M \subseteq L \subseteq K \subseteq J \subseteq (\mathbb{Z}/n\mathbb{Z})^\times$$

wie in [Bu].

Behauptung. Es gilt tatsächlich $K \subseteq J$.

Begründung. Zunächst folgt aus der Definition von m, dass m ein Teiler von $n-1$ ist. Genauer gilt $n - 1 = 2^{s-k}m$, und $s - k > 0$. *Das ist die Information, die ich bei unserem Gespräch übersehen hatte; es ist hier wichtig, dass d und s von vorneherein in Abhängigkeit von n definiert werden und die einzigen "Kandidaten" für die Zahl m die Zahlen d, 2d, …, $2^{s-1}d$ sind. In der Tat sind dies ja die Potenzen von Elementen in $(\mathbb{Z}/n\mathbb{Z})^\times$, die man für den Miller-Rabin-Test berechnen würde.*

Für $a \in K$ gilt also (für jede Primzahl p mit $p \mid n$)

$$a^{n-1} = (a^m)^{2^{s-k}} \equiv (\pm 1)^{2^{s-k}} = 1 \mod p^{e(p)}.$$

Deshalb genügt es, das folgende Lemma zu zeigen (das wir dann auf a^{n-1} und 1 anwenden):

Lemma 5. *Sind $a, a' \in (\mathbb{Z}/n\mathbb{Z})^\times$ und gilt $a \equiv a' \mod p^{e(p)}$ für alle Primzahlen p, die n teilen, dann folgt $a \equiv a' \mod n$.*

Beweis. In Vortrag 3 beweisen wir (im Beweis des im Seminarprogramm angegebenen Satzes), dass für teilerfremde Zahlen m_1 und m_2 die Abbildung

$$(\mathbb{Z}/(m_1 m_2)\mathbb{Z})^\times \to (\mathbb{Z}/m_1\mathbb{Z})^\times \times (\mathbb{Z}/m_2\mathbb{Z})^\times, \quad [a]_{m_1 m_2} \mapsto ([a]_{m_1}, [a]_{m_2})$$

bijektiv ist. (Hier schreiben wir in den Index an die eckigen Klammern, in welcher Gruppe wir die Restklasse von $a \in \mathbb{Z}$ gerade betrachten.)

Durch mehrfache Anwendung kann man das verallgemeinern zu der Aussage, dass die Abbildung

$$(1) \qquad \pi\colon (\mathbb{Z}/n\mathbb{Z})^\times \to (\mathbb{Z}/p_1^{e_1}\mathbb{Z})^\times \times \cdots \times (\mathbb{Z}/p_\ell^{e_\ell}\mathbb{Z})^\times, \quad [a]_n \mapsto ([a]_{p_1^{e_1}}, \ldots, [a]_{p_\ell^{e_\ell}})$$

bijektiv ist. Hier schreiben wir die Primfaktorzerlegung von n als $n = p_1^{e_1} \cdots p_\ell^{e_\ell}$ mit paarweise verschiedenen Primzahlen p_i.

Dass a und a' die Voraussetzung im Lemma erfüllen, bedeutet genau, dass $\pi([a]_n) = \pi([a']_n)$ ist. Wegen der Injektivität von π folgt dann $[a]_n = [a']_n$, wie gewünscht. □

Wir haben also die obige Kette von Untergruppen. Das Ziel ist, am Ende zu zeigen, dass $[(\mathbb{Z}/n\mathbb{Z})^\times : L] \geq 4$ gilt.

Lemma 6. (1) *Der Index $[K : M]$ ist eine Potenz von 2.*
(2) *Der Index $[K : L]$ ist ebenfalls eine Potenz von 2, und wir schreiben $[K : L] = 2^j$.*
(3) *Ist j wie in Teil (2), dann ist die Zahl $j+1$ genau die Anzahl der verschiedenen Primzahlen, die n teilen. (Mit der obigen Notation ist also $j + 1 = \ell$.)*

Beweis. Zu (1). Wir betrachten die folgende Kette von Untergruppen von $(\mathbb{Z}/n\mathbb{Z})^\times$:

$$M = K_\ell \subseteq K_{\ell-1} \subseteq K_{\ell-2} \subseteq \cdots \subseteq K_1 \subseteq K_0 = K,$$

wobei wir definieren:

$$K_t = \{[a] \in K;\ a^m \equiv 1 \mod p_i^{e_i} \text{ für alle } i = 1, \ldots, t\}.$$

(Dass $M = K_\ell$ gilt, folgt aus dem vorherigen Lemma.)

Behauptung. Für alle $t = 0, \ldots, \ell - 1$ gilt $[K_t : K_{t+1}] = 2$.

Begründung. Sei t fixiert. Wir müssen (nach Definition des Index) zeigen, dass es in K_t genau 2 Nebenklassen bezüglich der Untergruppe K_{t+1} gibt. Wir zeigen dazu dass $K_{t+1} \neq K_t$ gilt (also gibt es mindestens 2 Nebenklassen), und dass alle Elemente des Komplements $K_t \setminus K_{t+1}$ in derselben Nebenklasse liegen.

Um zu zeigen, dass $K_{t+1} \neq K_t$ gilt, genügt es, ein Element a in $(\mathbb{Z}/n\mathbb{Z})^\times$ finden, für das

$$a^m \equiv 1 \mod p_i^{e_i} \text{ für alle } i \neq t+1, \quad a^m \equiv -1 \mod p_{t+1}^{e_{t+1}}$$

gilt, denn dann liegt $[a]$ in K_t, aber nicht in K_{t+1}. Angesichts der Bijektion (1) oben können wir dieses Element ebenso gut im Produkt $(\mathbb{Z}/p_1^{e_1}\mathbb{Z})^\times \times \cdots \times (\mathbb{Z}/p_\ell^{e_\ell}\mathbb{Z})^\times$ angeben (und auch die geforderten Eigenschaften dort überprüfen, weil die Bijektion mit der Multiplikation von Restklassen kompatibel ist). Das bedeutet, dass es genügt, ganze Zahlen $a_1, \ldots, a_\ell$ zu finden, so dass

$$([a_1^m]_{p_1^{e_1}}, \ldots, [a_\ell^m]_{p_\ell^{e_\ell}}) = (1, \ldots, 1, -1, 1, \ldots, 1)$$

im Produkt $(\mathbb{Z}/p_1^{e_1}\mathbb{Z})^\times \times \cdots \times (\mathbb{Z}/p_\ell^{e_\ell}\mathbb{Z})^\times$ gilt (wobei rechts die -1 an der Stelle mit Index $t + 1$ steht).

Nach Definition von m existiert ein Element $b \in (\mathbb{Z}/n\mathbb{Z})^\times$ mit

$$b^m \equiv -1 \mod n$$

Erst recht gilt dann $b^m \equiv -1 \mod p_{t+1}^{e_{t+1}}$.

Wir können dann $a_i = 1$ für $i \neq t+1$ und $a_{t+1} = b$ setzen. Damit ist bewiesen, dass $K_{t+1} \neq K_t$ gilt.

Seien nun $[a], [a'] \in K_t \setminus K_{t+1}$. Um zu zeigen, dass die beiden Elemente in derselben Nebenklasse liegen, ist es nach dem zweiten Vortrag ausreichend zu zeigen, dass $[a]^{-1}[a'] \in K_{t+1}$ liegt. Das ist aber leicht nachzurechnen, denn

$$([a]^{-1}[a'])^m = [a^m]^{-1}[(a')^m] = [-1] \cdot [-1] = [1].$$

Aus der Behauptung folgt mit dem verallgemeinerten Satz von Euler-Lagrange (mehrfach angewandt), dass $[K : M] = 2^\ell$ gilt. Das beweist Teil (1).

Zu (2). Nach dem verallgemeinerten Satz von Euler-Lagrange gilt

$$[K : M] = [K : L] \cdot [L : M].$$

Deshalb folgt Teil (2) direkt aus Teil (1).

Zu (3). Wir haben in Teil (1) gesehen, dass $[K : M] = 2^\ell$ ist. Nach Definition von j gilt also

$$2^\ell = [K : M] = [K : L] \cdot [L : M] = 2^j \cdot [L : M].$$

Um den Beweis abzuschließen, genügt es nun zu zeigen, dass $[L : M] = 2$ gilt. Das Argument dafür ist ganz ähnlich zu dem, das wir schon in Teil (1) gesehen haben. Die Existenz eines Elements b mit $b^m \equiv -1 \mod n$ (wie in Teil (1) schon benutzt) zeigt auch, dass $M \neq L$, also $[L : M] > 1$ gilt. Nach Definition ist der Index $[L : M]$ die Anzahl der Nebenklassen von M in L. Eine der Nebenklassen ist jedenfalls M selbst. Es genügt nun also zu zeigen, dass alle Elemente von L, die nicht in M liegen, in derselben Nebenklasse von L liegen. Dafür genügt es zu zeigen, dass gilt:

$$[a], [a'] \in L \setminus M \quad \Longrightarrow \quad [a]^{-1}[a'] \in M.$$

Das ist aber klar, denn nach Definition von L und M folgt aus $[a], [a'] \in L \setminus M$, dass $a^m \equiv -1 \mod n$ und $(a')^m \equiv -1 \mod n$. Dann gilt aber $([a]^{-1}[a'])^m = ([a]^m)^{-1}[a']^m = [-1] \cdot [-1] = [1]$, also $[a]^{-1}[a'] \in M$. □

Insbesondere sehen wir mit diesem Lemma, dass im Fall $j = 1$ die Zahl n genau 2 Primfaktoren hat (und daher nach Übung 7.5 in [Bu] keine Carmichael-Zahl ist) und dass im Fall $j = 0$ die Zahl n höchstens einen Primfaktor hat, und damit dann eine Primzahlpotenz sein muss.

Zum Fall $j = 0$ (d.h. n ist eine Primzahlpotenz). In [Bu] wird behauptet, dass dann $\#J = p - 1$ gilt. Die Grundlage, das zu zeigen, ist die Tatsache, dass für Primzahlpotenzen $n = p^e$ die Gruppe $(\mathbb{Z}/n\mathbb{Z})^\times$ *zyklisch* ist, d.h. dass es ein Element $[c] \in (\mathbb{Z}/n\mathbb{Z})^\times$ gibt, so dass $(\mathbb{Z}/n\mathbb{Z})^\times$ gerade aus den Potenzen von $[c]$ besteht, also

$$(\mathbb{Z}/n\mathbb{Z})^\times = \{[c]^0 = [1], [c], [c]^2, \ldots, [c]^{\varphi(n)-1}\}.$$

Hier ist $\varphi(n) := \#(\mathbb{Z}/n\mathbb{Z})^\times$ wie in Vortrag 2. In der konkreten Situation gilt $\varphi(n) = \varphi(p^e) = (p-1)p^{e-1}$. Ein solches $[c]$ nennt man *Primitivwurzel (modulo n)*. Dass für eine Primzahlpotenz n immer eine Primitivwurzel existiert, wurde von Gauß bewiesen, siehe [Bs] Kapitel 2, §5.5.

Wenn man dieses Ergebnis benutzt, erhält man als Beschreibung der Untergruppe J die folgende:

$$J = \{[a] \in (\mathbb{Z}/n\mathbb{Z})^\times;\ [a]^{n-1} = [1]\} = \{[c]^i;\ [c]^{i(n-1)} = [1]\} = \{[c]^i;\ \varphi(n) \mid i(n-1)\}.$$

Nun ist $\varphi(n) = (p-1)p^{e-1}$ und $i(n-1) = i(p-1)(p^{e-1} + p^{e-2} + \cdots + p + 1)$ und daher gilt

$$\varphi(n) \mid i(n-1) \quad \Longleftrightarrow \quad p^{e-1} \mid i.$$

(Denn $p^{e-1} + p^{e-2} + \cdots + p + 1$ ist teilerfremd zu p.)

Also besteht J genau aus den Potenzen $[c]^i$ für die i ein Vielfaches von p^{e-1} ist und zwischen 0 und $(p-1)p^{e-1} - 1$ liegt, also

$$J = \{[c]^0 = [1], [c]^{p^{e-1}}, [c]^{2p^{e-1}}, \ldots, [c]^{(p-2)p^{e-1}}\}.$$

Folglich hat J genau $p-1$ Elemente, wie behauptet.

Zur Übungsaufgabe 7.5. Wenn man Theorem 7.4 aus [Bu] benutzt (und so ist die Übungsaufgabe dort ja sicher gedacht), dann ist es nicht so schwierig. Dann weiß man nämlich schon, dass in der Primfaktorzerlegung $n = p_1^{e_1} \cdots p_\ell^{e_\ell}$ mit Notation wie oben alle Exponenten $e_i = 1$ sind. Weil n keine Primzahl ist, ist nur noch der Fall $\ell = 2$ auszuschließen. Es ist dann zu zeigen, dass für $n = pq$ mit ungeraden Primzahlen $p \neq q$ nicht $p-1 \mid pq-1$ und $q-1 \mid pq-1$ gelten kann. Aber aus $p-1 \mid pq-1$ folgt auch $p-1 \mid pq-1-(p-1)q = q-1$, analog erhält man $q-1 \mid p-1$ und damit insgesamt $p-1 = q-1$, was wegen $p \neq q$ unmöglich ist.

(Den Beweis von Theorem 7.4 finde ich – speziell den Teil, wo gezeigt wird, dass n quadratfrei sein muss – allerdings auch ziemlich knapp.)

Vortrag 7 (Quadratisches Sieb)**.** Die Sicherheit von RSA ist nur unter der Annahme gewährleistet, dass es nicht "schnell" möglich ist, die Primfaktorzerlegung einer "großen" (zusammengesetzten) Zahl zu finden. In diesem Vortrag besprechen wir das Verfahren des quadratischen Siebs, das für große Zahlen wesentlich besser ist als der naive Ansatz, einfach die gegebene Zahl n per Division mit Rest durch $2, 3, \ldots$ zu teilen. (Wenn man RSA mit hinreichend großen Primzahlen durchführt (für die die notwendigen Rechnungen aber noch praktikabel sind), genügt das Verfahren aber nicht, um die Verschlüsselung zu knacken.)

Wir folgen dem Artikel [SW] von Schulz und Witten. Erklären Sie zunächste die Idee der Faktorisierung nach Fermat (siehe [SW] S. 71, 72); die Python-Implementierung und die genauen Laufzeitbetrachtungen sollten Sie weglassen, aber behandeln Sie das Beispiel, wie man $n = 1649$ mit dem Fermat-Algorithmus faktorisieren kann.

Erklären Sie dann das Faktorisieren von zusammengesetzten Zahlen mit dem *Quadratischen Sieb*, so wie in [SW] beschrieben. Siehe auch Wikipedia[12].

Ein Beispiel wäre nützlich (mindestens $n = 1649$, für das wir vorher schon das Fermat-Verfahren besprochen haben; vielleicht auch noch ein weiteres, allerdings scheint mir das Beispiel $n = 22\,213$ aus [SW] für den Vortrag zu aufwändig zu sein; es werden dort ja auch mehrere Quellen für andere Beispiele genannt – oder versuchen Sie, selbst eine geeignete Zahl zu finden).

[12] `https://de.wikipedia.org/wiki/Quadratisches_Sieb`

Auf das MPQS-Verfahren brauchen Sie nicht einzugehen.

Das quadratische Sieb wird auch bei Buchmann, [Bu] Abschnitt 9.3, besprochen.

Vortrag 8 (Der Algorithmus von Shor)**.** Der Vortrag hat zwei Teile. Wir beginnen mit:

(I) Der Algorithmus von Shor

Ziel des Algorithmus ist es, einen echten Teiler einer natürlichen Zahl $N \in \mathbb{N}_{>1}$ zu finden, die keine Primzahl ist (induktiv könnte man dann die Primfaktorzerlegung von N finden). Der naive Ansatz, dafür für alle Zahlen $2, 3, 4, \ldots$ die Division (mit Rest) von N durch diese Zahl durchzuführen, ist für große N nicht praktikabel. Man kennt zwar etwas bessere Methoden (wie das quadratische Sieb, das wir im vorherigen Vortrag besprochen haben; andere sind das Zahlkörpersieb und das Faktorisieren mit elliptischen Kurven), aber für „große“ Zahlen (mehr als 600 Stellen ist jedenfalls groß) dauern diese Verfahren in schlechten Fällen Jahre (oder vielleicht eher Jahrhunderte).

Die Grundidee des Algorithmus von Shor ist die folgende Version der Grundidee des Faktorisierens nach Fermat/mit dem quadratischen Sieb: Ist x eine zu N teilerfremde Zahl, $1 < x < N-1$, so dass $x^2 \equiv 1 \mod N$ gilt, dann haben wir $N \mid x^2 - 1 = (x-1)(x+1)$, aber $N \nmid x-1$, $N \nmid x+1$, und es folgt, dass $\mathrm{ggT}(N, x-1)$ ein echter Teiler von N ist (und ebenso $\mathrm{ggT}(N, x+1)$).

Um x zu finden, könnte man versuchen, zufällig eine Zahl a zwischen 1 und N zu wählen. Gilt $a^r \equiv 1 \mod N$ für eine *gerade* Zahl r, so wäre $a^{r/2}$ (bzw. genauer der Rest dieser Zahl modulo N) ein Kandidat für x. (Es bleibt allerdings zusätzlich zu prüfen, dass dieser Rest $\neq 1$ und $\neq N-1$ ist.)

Um das noch etwas besser zu formulieren, betrachten wir wieder die multiplikative Gruppe $(\mathbb{Z}/N\mathbb{Z})^\times$. Für $a \in (\mathbb{Z}/N\mathbb{Z})^\times$ sei $\mathrm{ord}(a)$ die Ordnung von a als Element von $(\mathbb{Z}/N\mathbb{Z})^\times$.

Wir suchen $a \in (\mathbb{Z}/N\mathbb{Z})^\times$, so dass $r := \mathrm{ord}(a)$ gerade ist und $a^{r/2} \neq -1$ (in $(\mathbb{Z}/N\mathbb{Z})^\times$).

Das führt auf die folgenden Schritte des Shor-Algorithmus:

Schritt 1. Wähle zufällig eine natürliche Zahl a, $1 < a < N$. Gilt $\mathrm{ggT}(a, N) > 1$, dann ist $\mathrm{ggT}(a, N)$ ein echter Teiler von N und wir haben das Ziel schon erreicht. Sonst sind a und N teilerfremd und die Restklasse von a ist ein Element von $(\mathbb{Z}/N\mathbb{Z})^\times$, das wir wieder mit a bezeichnen.

Schritt 2. Berechne $\mathrm{ord}(a)$. Wenn diese Zahl ungerade ist, müssen wir ein neues a wählen, d.h. wieder zu Schritt 1 zurückspringen.

Schritt 3. Wir haben a gefunden, so dass $r := \mathrm{ord}(a)$ gerade ist. Wenn $a^{r/2} = -1$ (in $(\mathbb{Z}/N\mathbb{Z})^\times$) gilt, müssen wir ein neues a wählen, d.h. wieder zu Schritt 1 zurückspringen. Sonst ist $\mathrm{ggT}(a^{r/2}+1, N)$ ein echter Teiler von N und wir sind fertig.

Die Hauptprobleme, die bleiben, sind die folgenden:

(1) Der Algorithmus basiert darauf, zufällig ein Element a zu wählen. Damit der Algorithmus in der Praxis sinnvoll anwendbar ist, muss die Wahrscheinlichkeit ein a zu wählen, das die im weiteren Verlauf benötigten Eigenschaften hat, genügend

groß sein. (Sonst müsste man eventuell sehr oft zu Schritt 1 zurückspringen und ein neues a wählen ...)

Man kann zeigen, dass diese Wahrscheinlichkeit (mindestens) $\frac{1}{2}$ ist, sofern N ungerade und keine Primzahlpotenz ist. Es ist klar, dass es genügt, ungerade N zu betrachten. Um festzustellen, ob N die Potenz einer Primzahl (und welcher Primzahl) ist, gibt es schnelle Verfahren, mit denen wir uns aber hier nicht beschäftigen. Für die verbleibenden N wird man dann in aller Regel nach wenigen Versuchen ein geeignetes a finden wird. Diese Abschätzung ist nicht sehr schwierig und auch nicht so interessant, und wir lassen sie deshalb aus. (Referenz, wenn Sie es genau wissen möchten: [Sch] Satz 6.7.)

Außerdem ist wichtig, dass für einen gegebenen Kandidat schnell überprüft werden kann, ob es sich tatsächlich um eine Lösung des gegebenen Problems handelt. Das ist hier der Fall, denn Multiplikation und Potenzieren in Restklassenringen ist „schnell“ möglich, ebenso der Test, ob eine gegebene Zahl die Zahl N teilt.

(2) Wie berechnet man in Schritt 2 die Ordnung von a? Es ist nicht klar, wie man das effizient machen kann, genauer ist kein Algorithmus bekannt, der diese Rechnung auf einem „klassischen“ Computer für große N in annehmbarer Zeit durchführt. Der wesentliche Punkt des Algorithmus von Shor ist, dass es auf *Quantencomputern* einen Algorithmus gibt, der dieses Problem „schnell“ lösen kann.

Literatur: [Sch], [KSV], [Ak], Wikipedia[13].

(II) Als zweiten Teil des Vortrags behandeln wir noch ein Lemma, das wir später bei der Durchführung des Algorithmus benötigen werden, und mit dessen Aussage wir uns in den kommenden beiden Vorträgen noch genauer befassen werden.

Lemma. Sei $x \in \mathbb{R}$ eine reelle Zahl und sei $s \in \mathbb{N}_{>0}$. Dann existiert höchstens eine Zahl $\xi \in \mathbb{Q}$, die sich mit Nenner $\in \{1, \dots, s\}$ schreiben lässt und für die $|x - \xi| < \frac{1}{2s^2}$ gilt.

Beweis. Wenn $\xi_1, \xi_2 \in \mathbb{Q}$ beide die genannten Bedingungen erfüllen, dann folgt $|\xi_1 - \xi_2| < 1/s^2$. Schreiben wir $\xi_i = \frac{a_i}{s_i}$ mit $a_i, s_i \in \mathbb{Z}$, $0 < s_1, s_2 \leq s$, dann haben wir andererseits $|a_1/s_1 - a_2/s_2| = |\frac{a_1 s_2 - a_2 s_1}{s_1 s_2}| \geq 1/s^2$, es sei denn es ist $a_1 s_2 - a_2 s_1 = 0$. Das würde aber gerade $\xi_1 = \xi_2$ bedeuten. □

Vortrag 9 (Kettenbrüche)**.** In diesem und dem folgenden Vortrag wollen wir zeigen, wie sich die Zahl ξ aus dem Lemma am Ende von Vortrag 8 explizit algorithmisch bestimmen lässt. Dazu besprechen wir einige Begriffe aus der Theorie der Kettenbrüche.

Literatur: Besonders gut gefällt mir, jedenfalls für unsere Zwecke, die Darstellung in [HW] Chapter 10. Andere Quellen, die teilweise ausführlicher sind, sind [Sa], [Sp].

Definieren Sie den Begriff des (regulären) Kettenbruchs $[a_0; a_1, a_2, \dots, a_n]$.

Zeigen Sie: $[a_0; a_1, \dots, a_m, [b_1, \dots, b_n]] = [a_0; a_1, \dots, a_m, b_1, \dots, b_n]$.

[13] `https://de.wikipedia.org/wiki/Shor-Algorithmus`

Definieren Sie die Konvergenten eines Kettenbruchs $[a_0; a_1, \dots]$. Genauer gesagt, definieren Sie

$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2} \quad (i \geq 2),$$
$$q_0 = 1, \quad q_1 = a_1, \quad q_i = a_i q_{i-1} + q_{i-2} \quad (i \geq 2),$$

und zeigen Sie:

(1) $\frac{p_i}{q_i} = [a_0; a_1, \dots, a_i]$ (*)
(2) $p_i q_{i-1} - p_{i-1} q_i = (-1)^{i-1}$.
(3) $p_i q_{i-2} - p_{i-2} q_i = (-1)^i a_i$.

Bis zu diesem Zeitpunkt kann man die a_i als "Variablen" betrachten (und entsprechend die p_i und q_i als Polynome); insbesondere gelten diese Formeln für beliebige reelle Zahlen a_i, sofern keiner der auftretenden Nenner $= 0$ wird. Dieser Standpunkt ist auch später noch einmal nützlich. Zunächst spezialisieren wir jetzt aber auf den Fall, dass alle a_i in $\mathbb{Z}$ liegen, und dass $a_i > 0$ für alle $i > 0$ (aber nicht zwingend für $i = 0$) gilt.

Zeigen Sie dann:

(1) Es gilt $\dots < \frac{p_{2i-2}}{q_{2i-2}} < \frac{p_{2i}}{q_{2i}} < \dots < \frac{p_{2i+1}}{q_{2i+1}} < \frac{p_{2i-1}}{q_{2i-1}} < \dots$,
(2) für alle $i \geq 1$ gilt $q_i \geq q_{i-1}$, und für $i > 1$ sogar $q_i > q_{i-1}$,
(3) für alle $i \geq 0$ sind p_i und q_i teilerfremd, die Darstellung in (*) ist also ein gekürzter Bruch.

Zeigen Sie, dass jede rationale Zahl als Kettenbruch geschrieben werden kann (indem Sie erklären, wie man das konkret macht, allgemein und an einem Beispiel) und besprechen Sie, inwiefern die Darstellungs eindeutig ist. Erklären Sie (aber ohne ausführliche Beweise) die entsprechenden Aussagen für nicht-rationale reelle Zahlen.

Diskutieren Sie an Beispielen, wie der Formalismus der Kettenbrüche "funktioniert". Besprechen Sie die Kettenbruchentwicklung der Zahl $\sqrt{2}$ und eventuell die der Kreiszahl π und/oder der Eulerschen Zahl e).

Vortrag 10 (Beste Näherungen durch rationale Zahlen)**.** In diesem Vortrag soll der Zusammenhang zwischen dem Lemma am Ende von Vortrag 8 und der Theorie der Kettenbrüche aus dem vorherigen Vortrag geklärt werden. Das zentrale Ergebnis ist der folgende Satz.

Satz. Sei $x \in \mathbb{R}$ eine reelle Zahl mit Kettenbruchentwicklung $[b_0; b_1, b_2, \dots]$. Sei $s \in \mathbb{N}_{>0}$ und sei $\xi \in \mathbb{Q}$ eine rationale Zahl, die sich mit Nenner $\in \{1, \dots, s\}$ schreiben lässt und für die $|x - \xi| < \frac{1}{2s^2}$ gilt.

Dann ist ξ eine der Konvergenten der Kettenbruchentwicklung von x, d.h. es existiert i mit $\xi = [b_0; b_1, b_2, \dots, b_i]$.

(Im Fall $x \in \mathbb{Q}$ (und das ist der Fall, in dem wir den Satz benutzen möchten) ist die Aussage natürlich so zu verstehen, dass i höchstens gleich dem letzten in der Kettenbruchentwicklung auftretendem Index ist.)

Beweis. Wir folgen [HW] Theorem 184. Für einen etwas anderen Ansatz siehe [Sa] Satz 2.28, [Sp] Satz 2.3.6. Wir schreiben $\xi = \frac{p}{q} = [a_0; a_1, \dots, a_n]$ mit $p, q \in \mathbb{Z}$ teilerfremd, $q > 0$, und so dass

$$\frac{p}{q} - x = (-1)^{n-1} \frac{\vartheta}{q^2}$$

für ein $\vartheta \in \mathbb{R}$, $0 \leq \vartheta < \frac{1}{2}$. (Überlegen Sie sich, warum das möglich ist und warum im Fall $\vartheta = 0$ nichts weiter zu tun ist, so dass wir nun $0 < \vartheta < \frac{1}{2}$ annehmen können.)

Seien $\frac{p_i}{q_i}$ die Konvergenten der Kettenbruchentwicklung von ξ. Dann existiert (warum?) ω mit

$$x = \frac{\omega p_n + p_{n-1}}{\omega q_n + q_{n-1}}.$$

Zeigen Sie, dass $\omega > 1$ gilt. Mit dem folgenden Lemma ergibt sich dann, dass $\xi = \frac{p_n}{q_n}$ eine Konvergente der Kettenbruchentwicklung von x ist. □

Lemma. Seien $p, q, r, s \in \mathbb{Z}$ mit[14] $q \geq s > 0$ und $\det \begin{pmatrix} p & r \\ q & s \end{pmatrix} \in \{1, -1\}$ und sei $\omega \in \mathbb{R}_{>1}$. Sei

$$x = \frac{\omega p + r}{\omega q + s}.$$

Dann sind $\frac{r}{s}$ und $\frac{p}{q}$ aufeinander folgende Konvergenten der Kettenbruchentwicklung von x.

Ist $\frac{p}{q} = [a_0; a_1, \dots, a_n]$, dann ist $x = [a_0; a_1, \dots, a_n, \omega]$.

Beweis. Siehe auch [HW] Theorem 172. Wir schreiben $\frac{p}{q} = [a_0; a_1, \dots, a_n]$ mit n so dass $ps - qr = (-1)^{n-1}$. Seien $\frac{p_i}{q_i}$ die Konvergenten dieser Kettenbruchentwicklung. Es folgt dann (überlegen Sie sich, oder schauen Sie nach, warum) $p = p_n$, $q = q_n$, $s = q_{n-1}$, $r = p_{n-1}$ und damit

$$x = \frac{\omega p_n + p_{n-1}}{\omega q_n + q_{n-1}} = [a_0; a_1, \dots, a_n, \omega].$$

(Die zweite Gleichheit folgt aus (*) oben.) Weil $\omega > 1$ gilt, können wir $\omega = [a_{n+1}, a_{n+2}, \dots]$ mit $a_{n+1} \geq 1$ schreiben und erhalten $x = [a_0; a_1, \dots, a_n, a_{n+1}, \dots]$. Damit sind alle Behauptungen des Lemmas bewiesen. □

Besprechen Sie als "Anwendung" von Kettenbrüchen, wie der Umgang mit Schaltjahren in verschiedenen Kalendersystemen mit der Kettenbruchentwicklung der "korrekten" Dauer eines Jahres im astronomischen Sinne (tropisches Jahr[15]) zusammenhängt ([Sp] 2.3, S. 32).

Wenn Zeit bleibt, zeigen Sie, dass für jede Konvergente $\frac{p_i}{q_i}$ der Kettenbruchentwicklung einer Zahl x gilt, dass ([HW] Theorem 164, [Sa] Satz 2.11, [Sp] Korollar 2.2.9)

$$\left| x - \frac{p_i}{q_i} \right| < \frac{1}{q_i q_{i+1}},$$

[14] In [**?**] wird $q > s$ gefordert, aber das ist für den Beweis nicht erforderlich und wir wollen, wenn wir das Lemma anwenden, den Fall $q = s$ zulassen. (In diesem Fall ist notwendigerweise $q = s = 1$ wegen der Determinantenbedingung.)

[15] `https://de.wikipedia.org/wiki/Tropisches_Jahr`

und eventuell die Verschärfung in [HW] Theorem 183.

Vortrag 11 (Komplexe Zahlen, Exponentialfunktion, Einheitswurzeln)**.** Erinnern Sie kurz an den Körper der komplexen Zahlen (Addition, Multiplikation, komplexe Konjugation, inklusive der "geometrischen" Interpretation in der komplexen Zahlenebene).

Erinnern Sie an die komplexe Exponentialfunktion und daran, wie Sinus- und Kosinus-Funktion damit zusammenhängen. Siehe zum Beispiel [Fo] §13, §14, [Kö] Abschnitte 3.1, 3.2, Kapitel 8, oder andere Lehrbücher der Analysis.

Beweisen Sie das folgende Lemma, das wir im letzten Vortrag verwenden werden.

Lemma.

(1) Für alle $\vartheta \in \mathbb{R}$ gilt $|e^{i\vartheta} - 1| = 2|\sin(\vartheta/2)|$.

(2) Für $\vartheta \in [0, \frac{\pi}{2}]$ gilt $\frac{2}{\pi}\vartheta \leq \sin(\vartheta) \leq \vartheta$.

(3) Für $\vartheta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ gilt $\frac{4}{\pi^2}\vartheta^2 \leq \sin(\vartheta)^2 \leq \vartheta^2$.

Beweis. Teil (1) folgt aus der Beziehung zwischen Sinusfunktion und komplexer Exponentialfunktion:

$$\sin(\alpha) = \frac{1}{2i}(e^{i\alpha} - e^{-i\alpha}).$$

Mit $\alpha = \vartheta/2$ bekommen wir

$$|\sin(\vartheta/2)| = \left|\frac{1}{2i}(e^{i\alpha} - e^{-i\alpha})\right| = \frac{1}{2}|e^{-i\alpha}(e^{2i\alpha} - 1)| = \frac{1}{2}|e^{i\vartheta} - 1|,$$

weil $|e^{-i\alpha}| = 1$ ist.

Teil (2) besagt, dass der Graph der Sinusfunktion im Intervall $[0, \pi/2]$ zwischen den Geraden mit Steigung $\frac{2}{\pi}$ und mit Steigung 1 liegt. Machen Sie sich das an einer Skizze klar. Welchen „Bezug" haben diese beiden Geraden zur Sinusfunktion?

Beide Aussagen kann man zeigen, indem man die Funktion $\sin(\vartheta) - 2\vartheta/\pi$ (bzw. $\sin(\theta) - \theta$) auf ihre Extrema im gegebenen Intervall untersucht ("Kurvendiskussion"). Die zweite Abschätzung kann man auch einfach mit Methoden der Integralrechnung beweisen, denn es gilt

$$\sin(\vartheta) = \int_0^\vartheta \cos(\tau)d\tau \leq \int_0^\vartheta 1 d\tau = \vartheta.$$

Teil (3) folgt direkt aus Teil (2), weil $\sin(-x) = -\sin(x)$ für alle $x \in \mathbb{R}$ gilt. □

Sei nun $Q \geq 1$ eine natürliche Zahl. Unter einer Q-ten Einheitswurzel in $\mathbb{C}$ verstehen wir eine komplexe Zahl $\omega \in \mathbb{C}$, für die $\omega^Q = 1$ gilt. Wir nennen ω eine *primitive Q-te Einheitswurzel*, wenn $\omega^Q = 1$, und $\omega^s \neq 1$ für alle $0 < s < Q$ gilt. Begründen Sie, dass die Q-ten Einheitswurzeln genau die komplexen Zahlen der Form $e^{\frac{2\pi ik}{Q}}$ mit $k = 0, \ldots, Q-1$ sind, und besprechen Sie, welche von diesen primitive Q-te Einheitswurzeln sind.

Erklären Sie die Lage der Q-ten Einheitswurzeln in der komplexen Zahlenebene.

Ist ω eine Q-te Einheitswurzel, dann können wir auch von den Potenzen ω^x für $x \in \mathbb{Z}/Q$ sprechen, denn jeder Repräsentant von x in $\mathbb{Z}$ liefert dasselbe Ergebnis.

Lemma. Sei $Q \in \mathbb{N}$ und sei $\omega = e^{2\pi i/Q} \in \mathbb{C}$. Sei $y \in \mathbb{Z}/Q$. Dann gilt die folgende Gleichung:

$$\sum_{x \in \mathbb{Z}/Q} \omega^{xy} = \begin{cases} Q & \text{wenn } y = 0, \\ 0 & \text{sonst.} \end{cases}$$

Beweis. Der erste Fall ist klar, denn $\omega^0 = 1$. Im zweiten Fall nutzen wir aus, dass die Menge der Summanden invariant ist unter der Drehung um $\frac{y}{Q} \cdot 360°$, also unter der Multiplikation mit ω^y:

$$\omega^y \sum_{x \in \mathbb{Z}/Q} \omega^{xy} = \sum_{x \in \mathbb{Z}/Q} (\omega^y)^{x+1} = \sum_{x \in \mathbb{Z}/Q} (\omega^y)^x.$$

Weil für $y \neq 0$ (in $\mathbb{Z}/Q$) die Potenz $\omega^y \neq 1$ ist, kann diese Gleichung nur gelten, wenn die Summe verschwindet.

(Im einfachsten Fall $Q = 2$ ist $\omega = -1$ und dann sind die obigen Gleichungen einfach $(-1)^0 + (-1)^0 = 2$ und $(-1)^0 + (-1)^1 = 0$.)

Alternativ kann man die Aussage mithilfe der endlichen geometrischen Reihe direkt nachrechnen. □

Vortrag 12 (Die unitäre Gruppe)**.** Erklären Sie zur Erinnerung die (aus der LA2 bekannte) Definition eines *unitären Vektorraums* (darunter verstehen wir einen endlich-dimensionalen $\mathbb{C}$-Vektorraum mit Skalarprodukt). Erläutern Sie das Beispiel des Standardvektorraums $\mathbb{C}^n$ mit dem Standardskalarprodukt. Definieren/diskutieren Sie den Begriff der Norm eines Vektors. Siehe zum Beispiel [Gö2], Abschnitt 19.4[16].

Definieren Sie den Begriff des unitären Endomorphismus ([Gö2] Definition 19.89, Satz 19.87) eines unitären Vektorraums. Die Menge aller unitären Abbildungen eines fixierten unitären Vektorraums V ist eine Gruppe unter der Verkettung von Abbildungen und heißt die *unitäre Gruppe von* V. Im Fall des Standardvektorraums $\mathbb{C}^n$ mit dem Standardskalarprodukt können wir Automorphismen als invertierbare Matrizen betrachten und erhalten die *unitäre Gruppe* $U(n)$, eine Untergruppe der $GL_n(\mathbb{C})$, siehe [Gö2] Definition 19.93, und den Begriff der unitären Gruppe zu einem unitären Vektorraum. Eine Matrix $A \in M_n(\mathbb{C})$ ist genau dann unitär, wenn $AA^* = E$ gilt, wobei A^* die Matrix ist, die aus A durch Transposition und Anwenden der komplexen Konjugation auf jeden Eintrag entsteht. Äquivalent ist auch, zu sagen, dass die Spalten von A eine Orthonormalbasis von $\mathbb{C}^n$ (bezüglich des Standardskalarprodukts) bilden.

Erklären Sie, dass für einen unitären Vektorraum V und jede lineare Abbildung $f\colon V \to V$ die folgenden Aussagen äquivalent sind ([Gö2] Satz 19.87):

(1) Die Abbildung f ist unitär.
(2) Für alle $v \in V$ mit $\|v\| = 1$ gilt $\|f(v)\| = 1$.

[16] `https://math.ug/la2-ss21/sec-blf-slf.html`

Beschreiben Sie die unitäre Gruppe $U(1)$.

Diskutieren Sie die folgenden Beispiele unitärer Matrizen:

(1) Jede Permutationsmatrix ist unitär.
(2) Sei $\omega \in \mathbb{C}$ eine n-te Einheitswurzel. Dann ist die Matrix $\frac{1}{\sqrt{n}}(\omega^{ij})_{i,j=0,\dots,n-1}$ unitär (benutzen Sie die Ergebnisse des vorhergehenden Vortrags).
(3) Punkt (2) liefert insbesondere: Für jedes $n \in \mathbb{N}$ gibt es eine unitäre Matrix $U \in M_n(\mathbb{C})$, für die alle Einträge der erste Spalten gleich (und zwar $= 1/\sqrt{n}$) sind. In dem Fall, dass n eine Zweierpotenz 2^s, $s \geq 1$, ist, kann man so eine Matrix alternativ aus einer Hadamard-Matrix[17] konstruieren.

Vortrag 13 (Quantencomputer 1)**.** Bevor wir beschreiben, was wir unter einem Quantencomputer verstehen, betrachten wir einen „klassischen“ Computer (bzw. ein „mathematisches Modell“ davon). Ein klassischer Computer besteht (für uns) aus seinem Speicher und einem Programm, das der Computer ausführen kann. Alle Daten im Speicher zusammen betrachten wir als ein s-Tupel von Nullen und Einsen, also als ein Element von $\{0,1\}^s$. Ein solches Tupel gibt den Zustand des Speichers zu einem Zeitpunkt an. Wir sagen, der Computer könne s Bits abspeichern, jedes Bit entspricht einem Eintrag des gegebenen s-Tupels von Nullen und Einsen. Zum Beispiel würden 16GB Speicherplatz gerade $s = 2^{37}$ Bits entsprechen. Die Menge $\{0,1\}^s$ hat in diesem Fall $2^{(2^{37})}$ Elemente, eine riesige Zahl. Wir nummerieren diese einzelnen Bits von 0 bis $s-1$ und schreiben $a_i \in \{0,1\}$ für den Inhalt/Zustand des i-ten Bits zu einem gegebenen Zeitpunkt ($i = 0, \dots, s-1$).

Das Programm besteht aus einer Folge von Operationen, wobei die einzelnen erlaubten Schritte die folgenden sind (für $i, j, k \in \{0, \dots, s-1\}$):

- Schreibe eine 0 oder eine 1 in Eintrag i,
- kopiere den Eintrag a_i an die Stelle j,
- ersetze den Inhalt von Bit i durch `NOT` $a_i := 1 - a_i$,
- ersetze den Inhalt von Bit i durch a_j `AND` $a_k := a_j \cdot a_k$,
- ersetze den Inhalt von Bit i durch a_j `OR` $a_k := 1 - (1 - a_j)(1 - a_k)$.

Entscheidend sind die letzten drei *logischen Operationen*. Auf der rechten Seite sind diese jeweils durch Rechnungen im Körper $\mathbb{F}_2$ ausgedrückt (was aber vielleicht nicht sehr hilfreich ist, machen Sie sich klar, warum diese Operationen `NOT`, `AND` und `OR` heißen).

Schreiben Sie ein Programm (im obigen Sinne), das zwei Zahlen addiert. Sie dürfen annehmen, dass die beiden Zahlen im Binärsystem im Speicher abgelegt sind, dass die erste Zahl zwei Stellen und die zweite Zahl eine Stelle hat. Das soll glaubhaft machen, dass man durch (sehr) viele solcher Schritte auch komplizierte Berechnungen durchführen kann, und zwar alle Rechnungen, die sich auf klassischen Computern eben durchführen lassen.

Den Computer kann man dann folgendermaßen benutzen: Man kann den Speicher „initialisieren“, also Nullen und Einsen in die Speicherplätze „eintragen“, das Programm ablaufen lassen, und dann den Speicher wieder auslesen.

[17] `https://de.wikipedia.org/wiki/Hadamard-Matrix`

Ein Quantencomputer hat auch einen „Speicher“ und kann ein „Programm abarbeiten“ und das Ergebnis „auslesen“. Wie wir sehen werden, sind Speicher und Programm wesentlich „besser“ als beim klassischen Computer; das Ergebnis auszulesen, ist aber „problematisch“, insbesondere *verändert das Auslesen des Ergebnisses in aller Regel den Inhalt des Speichers*!

Wir benötigen zur Beschreibung (eines Quantencomputers mit s Quantenbits) einen $\mathbb{C}$-Vektorraum V der Dimension 2^s mit einem Skalarprodukt. Wir können (und werden) einfach den Standardvektorraum $V := \mathbb{C}^{2^s}$ mit dem Standardskalarprodukt betrachten. Wir folgen der üblichen Notation und benennen die Standardbasisvektoren $e_1, \ldots, e_{2^s}$ folgendermaßen („ket-Notation“ aus der Physik):

$$|x\rangle, \quad x = 0, \ldots, 2^s - 1.$$

Oft schreibt man die natürliche Zahl x an dieser Stelle in Binärdarstellung, also zum Beispiel $111_2 = 7$ und entsprechend $|111_2\rangle$ für den achten Standardbasisvektor. (In der Literatur wird fast immer auch der Index $-_2$ für die Binärdarstellung weggelassen.)

Achtung: $|0\rangle$ ist nicht der Nullvektor, sondern einer der Basisvektoren!

Die möglichen Speicherzustände unseres Quantencomputers sind dann die Vektoren in V der Länge 1. Zum Beispiel sind die Standardbasisvektoren $|x\rangle$ selbst mögliche Zustände („klassische Zustände“, denn diese sind in Bijektion zu den Speicherzuständen eines klassischen Computers mit s Bits), aber eben auch alle Linearkombinationen

$$\sum_{x=0}^{2^s-1} a_x|x\rangle \quad a_x \in \mathbb{C} \text{ mit } \sum_x |a_x|^2 = 1$$

(man spricht von *Überlagerungen* klassischer Zustände). Während ein klassischer Computer mit einem Speicherplatz von s Bits also „nur“ 2^s verschiedene Speicherzustände hat, hat schon ein Quantencomputer mit nur einem einzigen Quantenbit unendlich viele Speicherzustände, nämlich alle Vektoren der Länge 1 in $\mathbb{C}^2$.

Ein Programm des Quantencomputers besteht aus einer Folge von unitären Matrizen $U \in M_{2^s}(\mathbb{C})$. In dem entsprechenden Schritt des Programms wird der Speicherzustand $v \in V$ ersetzt durch Uv. (Wir ignorieren in unserer Darstellung ein wichtiges “Detail”: In der Praxis wird es nicht möglich sein, beliebige unitäre Matrizen zu verwenden, sondern man wird – ähnlich wie wir den klassischen Computer mit einer sehr kleinen Anzahl von Operationen modelliert haben – auch beim Quantencomputer nur eine Auswahl unitärer Matrizen “implementieren” können, und dann nur solche unitären Matrizen als Befehle verwenden können, die man aus diesen Grundbausteinen geeignet kombinieren kann. Stichworte hierzu sind *Quanten-Gatter* und *Quanten-Schaltkreis*.)

Das Analogon zum „Auslesen“ des Speichers ist für den Quantencomputer die „Durchführung einer Messung“. Es ist ein essenzieller Bestandteil der Quantentheorie, dass der Zustand eines Systems eine Angabe von Wahrscheinlichkeiten ist und dass eine Messung den Zustand des Systems verändern kann (weil „die Messung ein genaues Ergebnis, nicht nur eine Wahrscheinlichkeit, liefern muss und dadurch den Zustand des Systems auf das gelieferte Ergebnis festlegt“).

Konkret: Ist $v = \sum_{x=0}^{2^s-1} a_x|x\rangle$ der aktuelle Zustand unseres Quantencomputers, dann liefert eine Messung zufällig ein Element $y \in \{0, \ldots, 2^s - 1\}$, und zwar wird jedes

y mit Wahrscheinlichkeit $|a_y|^2$ „gemessen“. (Weil v Länge 1 hat, ist die Summe aller dieser Wahrscheinlichkeiten gleich Eins.) Der Koeffizient a_y ist nicht Teil des Messergebnisses. Danach befindet sich der Computer im Zustand $|y\rangle$ (und das bedeutet: fast die komplette Information über den vorherigen Zustand ist durch die Messung verloren gegangen).

Für den Umgang mit Quantencomputern ist daher ein wesentlicher Aspekt, dass man damit rechnen muss, nur mit einer gewissen Wahrscheinlichkeit ein „brauchbares“ Ergebnis zu erhalten. Wenn man „Pech gehabt“ hat, muss man die Rechnung wiederholen und „hoffen“, dass es beim nächsten Mal klappt. Bei der Entwicklung von Programmen/Algorithmen für Quantencomputer muss man daher sicherstellen, dass die Programme mindestens mit einer hinreichend hohen Wahrscheinlichkeit zum Erfolg führen. Außerdem muss es möglich sein, schnell zu überprüfen, ob das Ergebnis der Quantencomputer-Rechnung tatsächlich eine Lösung des gegebenen Problems liefert. (Das ist ähnlich zu der Wahl von a in Schritt 1 des Algorithmus von Shor, wo auch das Risiko besteht, zufällig ein a mit ungerader Ordnung zu wählen, mit dem wir dann im weiteren Verlauf nichts anfangen können.)

Im nächsten Vortrag werden wir der Übersichtlichkeit halber die folgende Variante für die Schreibweise verwenden: Hat man einen Quantencomputer mit $s+t$ Quantenbits, hat der zugehörige unitäre Vektorraum die Dimension $2^{s+t} = 2^s 2^t$. Für die Basisvektoren bedeutet das, dass jeder Vektor in der Form $|x\rangle$ für eine Zahl x geschrieben werden kann, die im Binärsystem $\leq s+t$ Stellen hat. Wir füllen gegebenenfalls vorne mit Nullen auf, so dass wir genau $s+t$ Stellen haben und schreiben dann $|x\rangle = |y\rangle|z\rangle$, wo y durch die linken s Stellen von x und z durch die rechten t Stellen von x (im Binärsystem) gegeben sei. Zwei ket-Symbole nebeneinander zu schreiben, bedeutet also einfach, die beiden Zahlen im Binärsystem mit der vorgegebenen Stellenanzahl zu schreiben und nebeneinander zu schreiben. Man nennt dann das linke ket-Symbol das *erste Register* und das rechte ket-Symbol das *zweite Register* des Quantencomputers. (Mathematisch kann man diese Konstruktion mit dem Tensorprodukt von (unitären) Vektorräumen beschreiben, aber damit müssen wir uns nicht befassen.)

Begründen Sie, dass man auf nur einem der Register operieren kann, d.h., mit der obigen Notation: Ist $A \in U(2^s)$, dann existiert $A' \in U(2^{s+t})$ so dass $A'(|x\rangle|y\rangle) = (A|x\rangle)|y\rangle$, und ähnlich, wenn man auf dem zweiten Register operieren möchte.

Vortrag 14 (Quantencomputer 2)**.** In diesem Vortrag sollen einige Operationen, die ein Quantencomputer durchführen kann, konkretisiert werden. Im zweiten Teil besprechen wir die ersten Operationen, die man durchführt, um Schritt 2 aus Shors Algorithmus auf einem Quantencomputer durchzuführen. Dies wird dann im folgenden Vortrag weitergeführt.

Wir betrachten einen Quantencomputer mit zwei Registern, mit s und t Quantenbits.

(1) Initialisierung. Wir haben in Vortrag 12 besprochen, dass es eine unitäre Matrix $U \in U(2^s)$ gibt, so dass alle Einträge der ersten Spalte von U gleich $\frac{1}{\sqrt{2^s}}$ sind. Wir können U auf das erste Register des Quantencomputers anwenden. Was ist der Effekt, wenn man mit dem Zustand $|0\rangle|0\rangle$ beginnt?

(2) Durchführung klassischer Berechnungen auf Quantencomputern. Die Zustände $|x\rangle$ eines Quantencomputers mit s Quantenbits, die durch einen einzigen Standardbasisvektor gegeben sind, entsprechen gerade den Zuständen eines klassischen Computers mit s Bits. Eine "klassische Berechnung" (also eine, die man auch auf einem gewöhnlichen Computer durchführen kann) können wir als Funktion betrachten, die jedem Zustand, der durch einen einzigen Standardbasisvektor $|x\rangle$ gegeben ist, einen Zustand $|f(x)\rangle$ zuordnet, der wieder durch einen einzigen Standardbasisvektor gegeben ist. Ist die Rechnung umkehrbar, induziert sie also eine Permutation der Standardbasisvektoren, dann ist klar, dass sie auch auf einem Quantencomputer durchgeführt werden kann, weil jede Permutationsmatrix unitär ist.

Sei nun f eine Funktion, die jedem Standardbasisvektor einen Standardbasisvektor zuordnet, die aber nicht unbeding bijektiv ist. Um f auf einem Quantencomputer zu berechnen, nehmen wir ein zweites Register mit derselben Quantenbitanzahl zur Hilfe und betrachten die Abbildung $|x\rangle|y\rangle \mapsto |x\rangle|f(x) \oplus y\rangle$, wobei $\oplus$ die Addition in $\mathbb{F}_2^s$ bezeichnet. Diese Abbildung ist (warum?) invertierbar, und wir können $f(x)$ ausrechnen, indem wir die Abbildung auf $|x\rangle|0\rangle$ anwenden. Weil das Ergebnis ein einziger Standardbasisvektor ist, liefert eine Messung dann notwendigerweise das Paar $(x, f(x))$.

Wenn Sie genügend Zeit haben, könnten Sie an dieser Stelle noch das "no-cloning theorem" besprechen, siehe [Ak] Satz 1.4.1.

Wir beginnen nun mit der Erklärung, wie Schritt 2 aus dem Algorithmus von Shor auf einem Quantencomputer durchgeführt werden kann.

Wir benutzen einen Quantencomputer mit $2q$ Quantenbits, wobei q so gewählt wird, dass für $Q = 2^q$ und die zu faktorisierende Zahl N gilt, dass $N^2 \leq Q < 2N^2$ ist. Für die klassischen Zustände (also die Standardbasisvektoren des zugrundeliegenden unitären Vektorraums) benutzen wir die Schreibweise $|x\rangle|y\rangle$ mit $x, y \in \{0, \ldots, 2^q - 1\}$, wir teilen also die $2q = q + q$ Quantenbits auf in „zwei Register" mit jeweils q Quantenbits.

Es sei $a \in (\mathbb{Z}/N)^\times$ eine zu N teilerfremde Zahl und $f\colon \{0, \ldots, Q-1\} \to \mathbb{C}$ die Funktion $x \mapsto a^x$. Um die Werte von f als komplexe Zahlen zu betrachten, identifizieren wir $\mathbb{Z}/N = \{0, 1, 2, \ldots, N-1\} \subseteq \mathbb{C}$, wir wählen für die Restklasse a^x also immer den eindeutig bestimmten Repräsentanten in $\{0, 1, 2, \ldots, N-1\}$. Die Funktion f ist periodisch mit Periode $r := \mathrm{ord}(a)$, d.h. es ist $f(x + r) = f(x)$ für alle $x \in \mathbb{Z}$, für ein uns unbekanntes r, das wir gerne ermitteln möchten.

Wir führen nun auf dem Quantencomputer die folgenden Schritte durch.

(A) Initialisierung des Quantencomputers. Wir initialisieren den Computer im Zustand

$$\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle|0\rangle.$$

(B) Berechnung der Potenzen a^x. In diesem Schritt berechnen wir alle Werte $f(x)$ mit der vorher beschriebenen Methode. Hier verstehen wir $f(x) \in (\mathbb{Z}/N)^\times \subset \mathbb{Z}/N = \{0, \ldots, N-1\} \subset \{0, \ldots, Q-1\}$ mit der oben erklärten Identifikation $\mathbb{Z}/N =$

$\{0, \ldots, N-1\}$ als Element von $\{0, \ldots, Q-1\}$. Wir wenden also eine (unitäre) Abbildung mit $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ (für $x = 0, \ldots, Q-1$) an.

Damit können wir auf dem Quantencomputer gleichzeitig a^x für alle x berechnen (Man spricht von Quantenparallelismus. Es ist allerdings zu beachten, dass das Ergebnis nicht damit vergleichbar ist damit, alle Werte a^x auf einem klassischen Computer auszurechnen, weil man den im Quantencomputer erreichten Zustand nur einmal ausmessen kann, so dass es unmöglich ist, die Information über alle a^x "herauszuziehen".) Unser Quantencomputer wird so in den Zustand

$$\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle|f(x)\rangle$$

versetzt.

Mit diesem Zustand arbeiten wir im kommenden Vortrag weiter.

Vortrag 15 (Die Quanten-Fourier-Transformation)**.** In diesem Vortrag soll erklärt werden, wie man mit der sogenannten Quanten-Fourier-Transformation den *Schritt 2* des Shor-Algorithmus (siehe Vorträge 8, 14) abschließen kann.

Wir haben keine Zeit, in das Thema *Fourieranalysis* wirklich einzusteigen, und begnügen uns mit den folgenden motivierenden Bemerkungen:

- Als ganz grobe Beschreibung können wir sagen, dass eine Fouriertransformation dazu dient, vorliegende Daten in anderer (oftmals nützlicherer) Form zu "organisieren", also sie geeignet zu transformieren, und zar insbesondere unter Berücksichtigung von Symmetrieeigenschaften und speziell Periodizität.
- Grundidee der klassischen Fouriertransformation ist es, eine Funktion als Summe von „symmetrischeren" Funktionen auszudrücken. Ein sehr einfaches aber prototypisches Beispiel ist das folgende: Eine Funktion $f\colon \mathbb{R} \to \mathbb{R}$ heißt *gerade*, wenn $f(-x) = f(x)$ für alle $x \in \mathbb{R}$ gilt, und *ungerade*, wenn $f(-x) = -f(x)$ für alle $x \in \mathbb{R}$ gilt. (Was bedeutet das geometrisch für den Funktionsgraphen von f?)

  Ist nun $f\colon \mathbb{R} \to \mathbb{R}$ irgendeine Funktion, dann ist f die Summe der geraden Funktion $f_g\colon \mathbb{R} \to \mathbb{R}$, $f_g(x) = \frac{1}{2}(f(x) + f(-x))$, und der ungeraden Funktion $f_u\colon \mathbb{R} \to \mathbb{R}$, $f_u(x) = \frac{1}{2}(f(x) - f(-x))$. (Rechnen Sie nach, dass f_g gerade, f_u ungerade und $f(x) = f_g(x) + f_u(x)$ für alle $x \in \mathbb{R}$ ist – vielleicht nicht im Vortrag, aber jedenfalls für sich selbst).
- Die Fourierreihe einer periodischen Funktion drückt die Funktion als „Überlagerung", also als Summe, von trigonometrischen Funktionen aus. Das sollte man sich so vorstellen, dass ein (periodisches) Signal in seine verschiedenen Frequenzen zerlegt wird. (Eventuell wurde in Ihrer Analysis-Vorlesung das Thema Fourier-Reihen behandelt, sonst siehe gegebenenfalls zum Beispiel [Fo] Kapitel 23. Wir brauchen diese Theorie aber unten nicht, die Bemerkung dient nur dazu, einen Kontext für die Theorie der Quanten-Fouriertransformation herzustellen.)

Insofern ist es plausibel, dass eine Variante der Fouriertransformation dazu dienen kann, die unbekannte Periode einer periodischen Funktion zu ermitteln. Wie das

genau geht (und warum es mit einem Quantencomputer „schnell“ durchgeführt werden kann), ist aber nicht offensichtlich und soll nun erklärt werden.

Wir kommen dafür zurück zu Schritt 2 in Shors Algorithmus. Wir “übernehmen” aus dem vorherigen Vortrag einen Quantencomputer mit $2q$ Quantenbits (wobei q so gewählt wird, dass für $Q = 2^q$ und die zu faktorisierende Zahl N gilt, dass $N^2 \leq Q < 2N^2$ ist) im Zustand

$$\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle|f(x)\rangle.$$

Wie zuvor teilen wir die $2q$ Quantenbits in 2 Register mit jeweils q Quantenbits auf. Die Funktion f ist eine periodische Funktion $\{0, \ldots, Q-1\} \to \{0, \ldots, Q-1\} \subset \mathbb{C}$, deren Periode r wir bestimmen möchten.

(C) Die Quanten-Fouriertransformation. Wir definieren $\omega = e^{2\pi i/Q}$. Wir wenden die folgende unitäre Abbildung an (diese nennt man die Quanten-Fouriertransformation),

$$\mathbb{C}^Q \longrightarrow \mathbb{C}^Q, \quad |x\rangle \mapsto \frac{1}{\sqrt{Q}} \sum_{y=0}^{Q-1} \omega^{xy} |y\rangle.$$

(Dass diese Abbildung unitär ist, haben wir in Vortrag 12 besprochen.) Und zwar wenden wir die Abbildung auf das linke Register an und erhalten damit als neuen Zustand:

$$\begin{aligned}\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle|f(x)\rangle \mapsto & \frac{1}{Q} \sum_{x=0}^{Q-1} \sum_{y=0}^{Q-1} \omega^{xy} |y\rangle|f(x)\rangle \\ = & \frac{1}{Q} \sum_{y=0}^{Q-1} \sum_{z \in \mathrm{Im}(f)} \left(\sum_{x,\ f(x)=z} \omega^{xy} \right) |y\rangle|z\rangle.\end{aligned}$$

(D) Messung. Wir führen nun eine Messung des Zustands unseres Quantencomputers durch und bekommen damit einen Messwert der Form $|y\rangle|z\rangle$, und zwar erhalten wir das Paar $(y, z) \in \{0, \ldots, Q-1\} \times \mathrm{Im}(f)$ mit Wahrscheinlichkeit

$$\Pr(y, z) = \left| \frac{1}{Q} \sum_{x,\ f(x)=z} \omega^{xy} \right|^2,$$

(Paare (y, z), für die $z \notin \mathrm{Im}(f)$ ist, können nicht als Messwert auftreten.)

Für den Wert z unserer Messung schreiben wir $z = f(x_0)$ mit minimalem x_0. Dann ist

$$f^{-1}(z) = \{x_0, x_0 + r, \ldots, x_0 + (m-1)r\}$$

für $m = \#f^{-1}(z) = \lfloor \frac{Q - x_0 - 1}{r} \rfloor + 1$. Damit können wir wie folgt weiterrechnen:

$$\Pr(y, z) = \frac{1}{Q^2} \left| \sum_{b=0}^{m-1} \omega^{(x_0 + br)y} \right|^2 = \frac{1}{Q^2} \left| \sum_{b=0}^{m-1} (\omega^{ry})^b \right|^2.$$

Wir sehen an dieser Darstellung schon, dass die Quanten-Fouriertransformation dem Standardbasisvektor $|y\rangle|z\rangle$ eine Wahrscheinlichkeit zuordnet, die von ry abhängt.

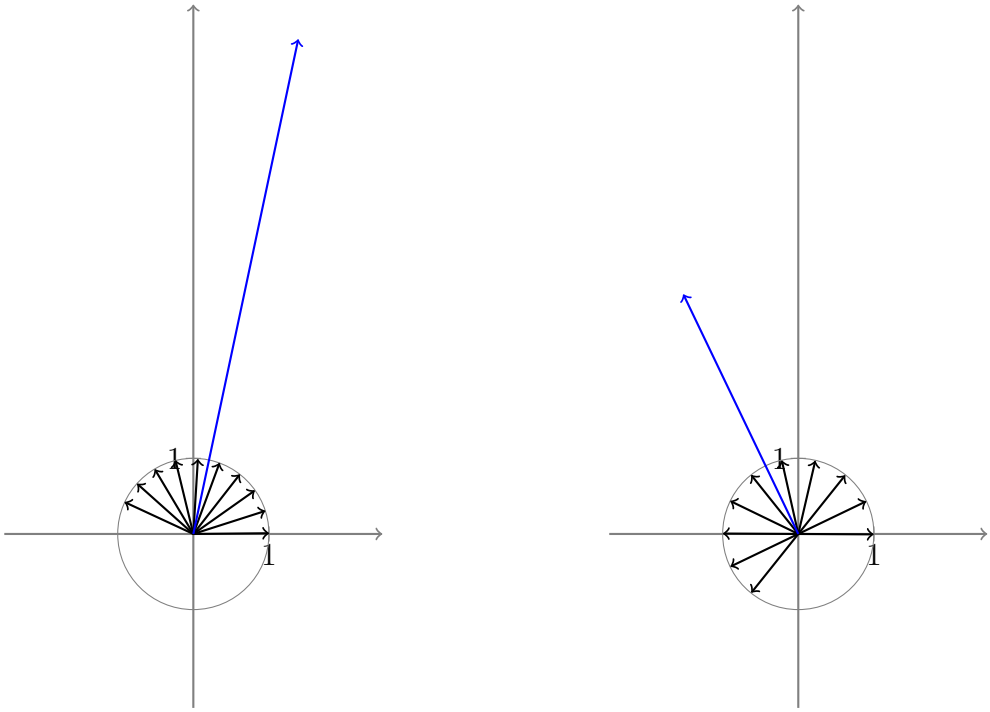
Genauer gilt: Wenn $r \mid Q$ gilt, ist ω^r eine primitive $\frac{Q}{r}$-te Einheitswurzel und die Summe auf der rechten Seite ist von der Form, über die wir in Vortrag 11 gesprochen haben, d.h. der Koeffizient von $|y\rangle|z\rangle$ ist genau dann von Null verschieden, wenn y ein Vielfaches von $\frac{Q}{r}$ ist. Durch eine Messung des Quantencomputer-Zustands findet man also als y ein Element der Form cQ/r von $\{0, \ldots, Q-1\}$. Wenn $c = 0$, also $y = 0$ ist, kann man mit dem Ergebnis nichts anfangen und muss erneut ansetzen. Sonst kann man $y/Q = c/r$ als gekürzten Bruch ausrechnen und erhält bestenfalls (wenn c und r teilerfremd sind) die Zahl r als den Nenner, oder andernfalls als Nenner immerhin einen nicht-trivialen Teiler von r. Das genügt normalerweise, um r zu bestimmen. Dieser Fall ist der Idealfall und illustriert die Grundidee, wie die Quanten-Fouriertransformation verwendet wird, um die Periode einer periodischen Funktion zu bestimmen. Wir lassen aber die weiteren Details des Falls $r \mid Q$ aus, weil er leichter zu behandeln und weniger relevant (immerhin ist Q eine Potenz von 2, so dass $r \mid Q$ nur gelten kann, wenn auch r eine Zweierpotenz ist) ist als der Fall $r \nmid Q$.

Wir nehmen also nun an, dass $r \nmid Q$ gilt. Dann sind wir nicht genau in der Situation der Vorüberlegung, aber – wie wir sehen werden – noch nah genug an dieser Situation, dass das gemessene Paar (y, z) es mit hoher Wahrscheinlichkeit ermöglicht, r zu berechnen.

Anschaulich können wir das wie folgt beschreiben: Mit hoher Wahrscheinlichkeit wird ein Wert (y, z) gemessen, für den $\Pr(y, z)$ groß ist, also $\left|\sum_{b=0}^{m-1} \omega^{byr}\right|$ groß ist. Damit diese Summe groß ist, müssen alle Summanden ω^{byr} möglichst eng beieinander auf dem Einheitskreis liegen. Das ist gerade dann der Fall, wenn ω^{ry} nahe bei 1 liegt, also ry sich nicht sehr von einem Vielfachen von Q unterscheidet, oder mit anderen Worten: Wenn der Bruch $\frac{yr}{Q}$ nahe an einer ganzen Zahl c liegt. Da y und Q bekannt sind, erhalten wir damit Informationen über r. (Wir werden das im folgenden Vortrag präzisieren.)

Die folgenden Abbildungen illustrieren das. Siehe auch das Applet dazu[18]. In beiden Fällen werden 10 Einheitswurzeln aufsummiert, die Summe ist durch den blauen Pfeil gegeben. Selbst wenn die Einheitswurzeln nur wenig mehr als einen Halbkreis des Einheitskreises „ausfüllen“ ist die Länge des Summenvektors deutlich kleiner als im anderen Fall. Genau diese Bedingung werden wir weiter unten wiedersehen und rechnerisch begründen.

[18] `https://math.ug/applets/sumen-einheitswurzeln.html`



Wie wir nachrechnen werden, ist $\frac{yr}{Q}$ mit hoher Wahrscheinlichkeit nah genug an einer ganzen Zahl, damit man aus der Kenntnis von y die Zahl r berechnen kann. Dass die Wahrscheinlichkeiten $\Pr(y,z)$ bei wenigen y konzentriert sind, sieht man auch, wenn man die Funktion $y \mapsto \sum_z \Pr(y,z)$ plottet. Siehe [Sh], Abb. 5.1 (was bei uns y heißt, heißt dort c) oder dieses Applet[19].

Um aus einem gemessenen Paar (y,z) in gewissen Fällen den gesuchten Wert der Periode r zu bestimmen, verwenden wir das Lemma aus Vortrag 8.

Sei c eine zu r teilerfremde Zahl zwischen 1 und r. Wegen der Teilerfremdheit können wir r bestimmen, sobald wir die rationale Zahl $\frac{c}{r}$ kennen. Wenn $y \in \{0,1,\dots,Q-1\}$ so beschaffen ist, dass $\frac{y}{Q}$ nahe an $\frac{c}{r}$ liegt, und zwar so dass $|\frac{y}{Q} - \frac{c}{r}| < \frac{1}{2N^2}$ gilt, dann folgt aus dem Lemma (mit $x := \frac{y}{Q}$, $s := N$), dass $\xi := \frac{c}{r}$ die *eindeutig bestimmte* rationale Zahl mit Nenner $\leq N$ ist, die diese Abschätzung erfüllt. Wie wir in Vortrag 10 gesehen haben, kann man die Zahl $\frac{c}{r}$ in effizienter Weise aus y, Q und N (die uns nach der Messung des Quantencomputer-Zustands alle bekannt sind) bestimmen, indem man die entsprechende Kettenbruchentwicklung berechnet.

Wir würden also (wenn wir einen Quantencomputer gebaut hätten) so vorgehen: Wir führen die obige Prozedur durch und messen y. Wir nähern y möglichst gut durch eine Bruchzahl mit kleinstem Nenner $r \leq N$ an und überprüfen dann, ob r tatsächlich die Ordnung des betrachteten Elements a ist. Wenn ja, ist Schritt 2 von Shors Algorithmus abgeschlossen. Wenn nein, starten wir einen erneuten Durchlauf des Quantencomputers. (In der Praxis ist es sinnvoll, eventuell noch kleine Vielfache von r daraufhin zu untersuchen, ob sie gleich $\mathrm{ord}(a)$ sind.)

[19] `https://math.ug/applets/plot-qft.html`

Die verbleibende Frage ist, mit welcher Wahrscheinlichkeit man damit rechnen kann, ein Paar (y, z) zu messen, dessen erster Eintrag y die obige Bedingung erfüllt und folglich das gesuchte r liefert. Damit beschäftigen wir uns im nächsten Vortrag.

Vortrag 16 (Berechnung von $\Pr(y, z)$)**.** Wir verwenden die Bezeichnungen aus dem vorherigen Vortrag. (Erinnern Sie zu Beginn des Vortrags noch einmal daran, wo wir stehen und was noch zu tun ist.)

Wir wollen zeigen, dass die Wahrscheinlichkeit, einen Eintrag (y, z) zu messen, für den

$$\left|\frac{y}{Q} - \frac{c}{r}\right| < \frac{1}{2Q} \quad \text{für ein zu } r \text{ teilerfremdes } c \in \{1, \dots, r\}$$

gilt, größer als (ungefähr) $1/(6\ln(\ln(N)))$ ist.

(Weil nach Wahl von Q gilt, dass $N^2 \leq Q$ ist, erfüllen diese y die Bedingung, die dann – wie zuvor diskutiert – die Bestimmung von r erlaubt.)

Der Wert $1/(6\ln(\ln(N)))$ erscheint auf den ersten Blick vielleicht zu klein, um nützlich zu sein. Aber da die Funktion $x \mapsto \ln(\ln(x))$ extrem langsam wächst, kann man damit trotzdem arbeiten. Man wird zwar damit rechnen müssen, den Algorithmus mehrfach durchführen zu müssen, aber wenn die Dauer der einzelnen Durchläufe im Sekunden- oder Minutenbereich ist, hat man trotzdem etwas gewonnen (sogar dann noch, wenn jeder Durchlauf wenige Stunden dauert ...). Dazu eine Überschlagsrechnung: Ist N ungefähr von der Größe 10^{1000} (also 1000 Stellen), so ist $\ln(\ln(N))$ etwas kleiner als 8, die Erfolgswahrscheinlichkeit ist also etwas besser als $\frac{1}{50}$. Mit 230 Versuchen hätte man damit schon eine Erfolgswahrscheinlichkeit von über 99%. Mit klassischen Computern würden die besten bekannten Methoden in schwierigen Fällen Jahre für die Faktorisierung einer so großen Zahl benötigen. (Siehe RSA challenge[20] für einige Beispiele dazu.)

In den folgenden Abschnitten leiten wir die genannte Abschätzung her.

Wir formen den Ausdruck, den wir für die einzelnen Wahrscheinlichkeiten $\Pr(y, z)$ erhalten haben, etwas weiter um. Mit der endlichen geometrischen Reihe und Teil (1) des Lemmas aus Vortrag 11 erhalten wir (wegen $\omega = e^{2\pi i/Q}$):

$$\Pr(y, z) = \frac{1}{Q^2}\left|\sum_{b=0}^{m-1}(\omega^{ry})^b\right|^2 = \frac{1}{Q^2}\frac{|\omega^{mry} - 1|^2}{|\omega^{ry} - 1|^2} = \frac{1}{Q^2}\frac{\sin(m\frac{ry}{Q}\pi)^2}{\sin(\frac{ry}{Q}\pi)^2}.$$

Die Funktion $x \mapsto \sin(x\pi)^2$ ist periodisch mit Periode 1, bei der Zahl $\frac{ry}{Q}$ im letzten Ausdruck kommt es also nur auf den gebrochenen Anteil an. Wir schreiben $\left\{\frac{ry}{Q}\right\}$ für die eindeutig bestimmte Zahl im Intervall $(-\frac{1}{2}, \frac{1}{2}]$, für die $\frac{ry}{Q} - \left\{\frac{ry}{Q}\right\} \in \mathbb{Z}$ gilt. Es gilt dann also

$$\Pr(y, z) = \frac{1}{Q^2}\frac{\sin(m\left\{\frac{ry}{Q}\right\}\pi)^2}{\sin(\left\{\frac{ry}{Q}\right\}\pi)^2}.$$

[20] `https://de.wikipedia.org/wiki/RSA_Factoring_Challenge`

Für die folgende Rechnung tun wir so, als ob $m = \frac{Q}{r}$ gelten würde. Da der Unterschied zwischen diesen beiden Zahlen sehr klein ist, hat das auf die weitere Rechnung keinen signifikanten Einfluss. Wir schreiben jeweils $\approx$, wenn diese Ersetzung vorgenommen wird, nehmen aber an, dass die entsprechenden Abschätzungen auch danach noch gelten.

Für $\left|\left\{\frac{ry}{Q}\right\}\right| \leq \frac{1}{2m} \approx \frac{r}{2Q}$ erhalten wir mit Teil (3) des Lemmas aus Vortrag 11:

$$\Pr(y, z) \geq \frac{4m^2}{\pi^2 Q^2} \approx \frac{4}{\pi^2 r^2}.$$

(Dies ist genau der Fall, dass in der Summe $\sum_{b=0}^{m-1} (\omega^r)^{by}$ alle Punkte ω^{rby} auf (höchstens) einem Halbkreis des Einheitskreises konzentriert sind.)

Jetzt können wir zusammenfassen: Sei $\varphi(r)$ wie in Vortrag 1 die Anzahl der zu r teilerfremden Zahlen zwischen 1 und r. Für jede solche Zahl c existiert (warum?) ein (eindeutig bestimmtes) $y \in \{0, \ldots, Q-1\}$ mit $|\frac{y}{Q} - \frac{c}{r}| \leq \frac{1}{2Q}$. Für dieses y gilt $\left\{\frac{ry}{Q}\right\} \leq \frac{r}{2Q} \approx \frac{1}{2m}$ und damit $\Pr(y, z) \geq \frac{4}{\pi^2 r^2}$ für jedes z von der Form a^x. Es gibt genau r solche Werte für z. Die Wahrscheinlichkeit, dieses y (mit irgendeinem z im zweiten Eintrag) zu messen ist folglich $\geq \frac{4}{\pi^2 r}$. Da es $\varphi(r)$ Möglichkeiten für c gibt, sehen wir, dass wir mit Wahrscheinlichkeit

$$\geq \frac{4}{\pi^2} \cdot \frac{\varphi(r)}{r}$$

ein y messen, „mit dem wir arbeiten können“, d.h. aus dem wir in der oben beschriebenen Weise r bestimmen können.

Wir verwenden nun den folgenden Satz aus der Zahlentheorie:

Satz.

$$\liminf_{r \to \infty} \frac{\varphi(r)}{r} \cdot \ln(\ln(r)) = e^{-\gamma} \approx 0,56$$

gilt. (Hier ist γ die sogenannte Eulersche γ-Konstante.)

Beweis. Der Satz ist nicht leicht zu beweisen. Man benötigt den *Primzahlsatz* über die Verteilung der Primzahlen (dessen Aussage können Sie vielleicht angeben, [HW] Theorem 6 in Abschnitt 1.8, Wikipedia[21]). Der Satz, den wir benötigen ist [HW] Theorem 328, Beweis in Abschnitt 22.9, aber der Beweis ist sicher nicht im Vortrag unterzubringen. □

Wenn wir $\frac{4}{\pi^2} \geq \frac{1}{3}$ und mit dem gerade genannten Resultat $\frac{\varphi(r)}{r}$ gegen $\frac{1}{2\ln(\ln(r))}$ (für genügend große r) abschätzen, bekommen wir das gewünschte Ergebnis.

[21] `https://de.wikipedia.org/wiki/Primzahlsatz`

Literaturverzeichnis

Auf die verlinkten Bücher sollten Sie kostenfrei über Ihren Uni-Account zugreifen können. Sie müssen sich dazu auf der Springer-Seite einloggen (klicken Sie *Login* auf der Springer-Seite und wählen Sie den Login über die Institution, d.h. die Universität Duisburg-Essen). Bitte melden Sie sich gegebenenfalls, wenn Sie Probleme haben, die Literatur, die Sie benötigen, zu beschaffen.

Die Texte, zu denen kein Link angegeben ist, finden Sie auf der Moodle-Seite. *Diese Dokumente sind nur für den Gebrauch für dieses Seminar bestimmt. Sie dürfen die Dateien nicht an andere Personen weitergeben oder anderweitig veröffentlichen.*

[Ak] B. Aktas, *Quantum computing und Shor-Algorithms*, Bachelor-Arbeit Univ. Ulm, 2019
https://oparu.uni-ulm.de/xmlui/handle/123456789/26053

[Bu] J. Buchmann, *Einführung in die Kryptographie*, 6. Aufl., Springer 2016,
https://doi.org/10.1007/978-3-642-39775-2

[Bs] P. Bundschuh, *Einführung in die Zahlentheorie*, 6. Aufl., Springer 2008.
https://doi.org/10.1007/978-3-540-76491-5

[Fo] O. Forster, *Analysis 1*, 12. Aufl. (oder eine andere Auflage), Springer Spektrum, 2016
https://doi.org/10.1007/978-3-658-11545-6

[Gö1] U. Görtz, *Lineare Algebra 1*, Vorlesungsskript WS 2020/21,
https://math.ug/la1-ws2021/

[Gö2] U. Görtz, *Lineare Algebra 2*, Vorlesungsskript SS 2021,
https://math.ug/la2-ss21/

[HW] G. Hardy, E. Wright, *An Introduction to the Theory of Numbers*, 6th ed., Oxford Univ. Press (2008).

[KSV] A. Kitaev, A. Shen, M. Vyalyi, *Classical and quantum computation*, Graduate Studies in Math. **47**, Amer. Math. Soc. (2002).

[Kö] K. Königsberger, *Analysis 1*, 6. Aufl., Springer (2004).
https://doi.org/10.1007/978-3-642-18490-1

[Ma] Y. Manin, *Classical computing, quantum computing, and Shor's factoring algorithm*, in Séminaire Bourbaki 1998/99, Astérisque **266**, Exposé **862** (2000).
http://www.numdam.org/item/SB_1998-1999__41__375_0/

[Sch] W. Scherer, *Mathematik der Quanteninformatik*, Springer 2016
https://doi.org/10.1007/978-3-662-49080-8

[SW] R.-H. Schulz, H. Witten, *Faktorisieren mit dem Quadratischen Sieb*, LOG IN **172/173** (2011/2012), 70–78.

[Sa] T. Sauer, *Kettenbrüche*, Vorlesungsskript,
https://www.fim.uni-passau.de/fileadmin/dokumente/fakultaeten/fim/lehrstuhl/sauer/geyer/Kettenbrueche.pdf

[Sp] J. Sprang, *Einführung in die transzendente Zahlentheorie*, Vorlesungsskript,
https://homepages.uni-regensburg.de/~spj54141/SS2020/Skript_final.pdf

[Sh] P. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, https://arxiv.org/pdf/quant-ph/9508027v2.pdf

Anmerkungen (Wie hat's geklappt?)

- Der Euklidische Algorithmus war nicht allgemein bekannt. (Ich hatte ihn "sicherheitshalber" im ersten Vortrag besprochen.)
- Vorträge 2 und 3 sind "unproblematisch", allerdings waren die Vorkenntnisse über allgemeine Gruppentheorie sehr dürftig (Nebenklassen, Quotient, Ordnung waren für viele einschließlich der Vortragenden neue (bzw. fast vollständig vergessene) Begriffe). Dadurch war insbesondere Vortrag 2 (und der erste Teil von Vortrag 3) schwieriger als Vortrag 7 über Kettenbrüche.
- Der letzte Teil von Vortrag 4 (Thm. 7.6 aus [Bu]) ist lang und im Vergleich schwierig (und benutzt mehr Gruppentheorie als vorher behandelt wurde). Aus diesem Vortrag könnte man gegebenenfalls ohne weiteres zwei Vorträge machen.
- Vortrag 5 hat gut funktioniert, ist aber mathematisch nicht so anspruchsvoll – es wird eigentlich nichts bewiesen.
- Vortrag 6 ist deutlich kürzer als die vorherigen. Man kann jedenfalls noch erklären, wie man Potenzen a^n für große n schnell berechnet (berechne a^{2^i} und benutze die Binärdarstellung von n). Eventuell könnte man auch Satz 6.7 aus [Sch] aufnehmen.
- Vorträge 7 und 8 sind prima (und für die Zielgruppe anscheinend leichter als die Vorträge 1 und 2).
- Vorträge 9 und 10 sind relativ leicht und locker in den angepeilten 80 Minuten zu schaffen.
- Vortrag 11 ist trotz der Länge der Beschreibung im Programm recht kurz und ich würde ihn beim nächsten mal "umorganisieren". Ähnliches gilt für Vortrag 12; dort wäre eine Option, das no-cloning-theorem in den Pflichtteil zu verschieben.
- Vortrag 13 ist gut in der vorgesehenen Zeit zu schaffen, und man kann noch die Applets per Beamer kurz vorführen. Vortrag 14 ist ebenfalls gut machbar (für eine Erklärung des Primzahlsatzes hatte ich aber keine Zeit mehr).

Insgesamt hat das Seminar gut geklappt und ist das Thema, wie ich denke, gut angekommen.