Mathematik für die Informatik II

Numerik und Diskrete Mathematik

Goethe-Universität Frankfurt am Main

Sommersemester 2018

Dr. Samuel Hetterich

10. April 2018

Inhaltsverzeichnis

Vorwort

Dieses Skript ist Grundlage der Vorlesung “Diskrete Mathematik und Numerik - Mathematik für die Informatik II” gehalten von Dr. Samuel Hetterich im Sommersemester 2017 an der Goethe-Universität in Frankfurt am Main.

Das Skript wird im Laufe des Semesters entwickelt - die entsprechenden Abschnitte sollten aber in ihrer endgültigen Form jeweils vor den einzelnen Vorlesungen zur Verfügung stehen. Bei Anmerkungen, Kritik und Korrekturvorschlägen zögern Sie bitte nicht, sich an Dr. Samuel Hetterich (hetterich@math.uni-frankfurt.de) zu wenden.

Teile des vorliegenden Manuskripts sind aus den Skripten zu der gleichen Vorlesung vorangegangener Semester von Herrn Dr. Hartwig Bosse übernommen.

Numerische Differentialgleichungen.

Nummerische Integrale.

Optimierungsprobleme Gradient descent Lagrange Multiplikatoren.

1 Grundlagen

Dieses Kapitel enthält Grundlagen aus der Vorlesung “Mathe für die Informatik I” und richtet sich an all jene, die diese Vorlesung noch nicht gehört haben.

Wir beginnen mit einigen sehr grundlegenden mathematischen Konzepten, die zum Teil schon aus der Schulmathematik bekannt sein sollten und in der Vorlesung häufig als “Handwerkszeug” in den unterschiedlichen Kontexten auftauchen werden.

1.1. Mathematische Logik: Aussagen und Logische Quantoren

Unter einer mathematischen Aussage versteht man eine mathematische Formel, oder eine formal-logische Aussage, der ein Wahrheitswert “wahr” oder “falsch” zugewiesen werden kann.

- Der Ausdruck “$x^2 - 2x + 1$” ist *keine* mathematische Aussage sondern nur ein mathematischer Term.
- Der Ausdruck “$x^2 - 2x + 1 = 0$” ist eine mathematische Aussage (die je nach Wert von x wahr oder falsch ist).
- Der Ausdruck “$1 = 0$” ist eine mathematische Aussage, die falsch ist.
- Der Ausdruck “4 ist eine Quadratzahl” ist eine mathematische Aussage, die richtig ist.
- Die Goldbach-Vermutung “Jede gerade natürliche Zahl größer als 2 kann als Summe zweier Primzahlen geschrieben werden.“ ist eine mathematische Aussage von der bisher nicht klar ist, ob sie wahr oder falsch ist.

Wie Aussagen im “normalen” Leben, muss jede mathematische Aussage ein Verb enthalten. Diese Verben stecken oft in *logischen Quantoren* oder *logischen Operatoren*, die im Grunde Abkürzungen für Textbausteine sind. In den obigen Beispielen steckt das Verb an einigen Stellen in dem Operator “=”, den man als “. . . ist gleich . . .” liest.

In den folgenden Tabelle sind die von uns verwendeten logischen Operatoren und Quantoren aufgelistet.

- **Liste der verwendeten Operatoren:**

Symbol	Name	Zugehörige Formulierung	Beispiel
$\neg$	Negation	Es gilt nicht . . .	$\neg[3 = 4]$
$\vee$	Oder	Es gilt . . . oder . . .	$[n \geq 2] \vee [n \leq 2]$
$\dot{\vee}$	Exklusives Oder	Es gilt **entweder** . . . oder . . .	$[n \geq 2] \dot{\vee} [n \leq 2]$
$\wedge$	Und	Es gilt . . . und . . .	$[n \geq 2] \wedge [n \leq 2]$

- **Liste der verwendeten Quantoren:**

Symbol	Name	Zugehörige Formulierung	Beispiel
$\forall$	All-Quantor	Für alle . . .	$\forall n \in \mathbb{N} : n \geq 0$
$\exists$	Existenz-Quantor	Es existiert (mindestens) ein . . .	$\exists n \in \mathbb{N} : n \geq 5$
$\exists!$		Es existiert **genau** ein . . .	$\exists! n \in \mathbb{N} : n^2 = 25$
$\nexists$		Es existiert **kein** . . .	$\nexists n \in \mathbb{N} : n < 0$

Es gelten in gewissem Sinne “Rechenregeln” für mathematische Aussagen. Dabei spielen die Begriffe der ***Äquivalenz*** und der ***Implikation*** eine entscheidende Rolle, welche mathematische Aussagen in Relation setzen.

Definition 1.1.1.

- Eine mathematische Aussage $\mathcal{A}$ ***impliziert*** eine weitere mathematische Aussage $\mathcal{B}$, wenn aus der Wahrheit der Aussage $\mathcal{A}$ die Wahrheit der Aussage $\mathcal{B}$ folgt. Man schreibt dann

$$\mathcal{A} \Rightarrow \mathcal{B}.$$

- Zwei mathematische Aussagen $\mathcal{A}$ und $\mathcal{B}$ sind ***äquivalent***, wenn $\mathcal{A}$ die Aussage $\mathcal{B}$ impliziert und umgekehrt auch $\mathcal{B}$ die Aussage $\mathcal{A}$ impliziert. In diesem Fall schreibt man

$$\mathcal{A} \Leftrightarrow \mathcal{B}.$$

(Die “Formel”: *. . . genau . . . dann . . . , wenn . . .* weist auf Äqivalenz in gesprochener Sprache hin.)

Beispiel 1.1.2.

- Es ist $n \geq 5 \Rightarrow n \geq 3$. Umgekehrt ist dies jedoch nicht der Fall.
- Ein Dreieck ist genau dann gleichseitig, wenn alle Seiten die gleiche Länge haben.

Bemerkung 1.1.3.

Interessanterweise sind die Implikationen $\mathcal{A} \Rightarrow \mathcal{B}$ und $\neg\mathcal{B} \Rightarrow \neg\mathcal{A}$ gleichbedeutend. Denn wenn $\mathcal{A}$ die Aussage $\mathcal{B}$ impliziert, dann kann $\mathcal{A}$ nicht wahr sein, wenn $\mathcal{B}$ nicht wahr ist. Ergo impliziert $\neg\mathcal{B}$ die Aussage $\neg\mathcal{A}$.

Bemerkung 1.1.4.

Im Fall der Äquivalenz sind die Aussagen entweder beide wahr oder beide falsch - sie sind gleichwertig. Daher erschließt sich der Name aus dem Lateinischen: *aequus* “gleich” und *valere* “wert sein”.

Eine schöne Veranschaulichung für den Unterschied zwischen Äquivalenz und Implikation ist diese Eselsbrücke, welche den Sachverhalt der Impliaktion veranschaulicht:

> *Wenn es geregnet hat, ist die Straße nass.*
> *Wenn die Straße nass ist, heißt das nicht zwangsläufig, dass es geregnet hat.*

$$\text{“Es hat geregnet.”} \Rightarrow \text{“Die Straße ist nass.”}$$

$$\text{“Die Straße ist nass.”} \not\Rightarrow \text{“Es hat geregnet.”}$$

Aber es ist nach Bemerkung 1.1.3

$$\neg(\text{“Die Straße ist nass.”}) \Rightarrow \neg(\text{“Es hat geregnet.”})$$

was umformuliert heißt

"Die Straße ist nicht nass." $\Rightarrow$ "Es hat nicht geregnet."

Wie angekündigt nun die "Rechenregeln" für mathematische Aussagen.

Lemma 1.1.5. Es seien $\mathcal{A}, \mathcal{B}, \mathcal{C}$ mathematische Aussagen. Dann gelten

$$\begin{aligned} \mathcal{A} \wedge \mathcal{B} &\Leftrightarrow \mathcal{B} \wedge \mathcal{A} \\ \mathcal{A} \vee \mathcal{B} &\Leftrightarrow \mathcal{B} \vee \mathcal{A} \end{aligned} \qquad \text{(Kommutativgesetze)}$$

$$\begin{aligned} [\mathcal{A} \wedge \mathcal{B}] \wedge \mathcal{C} &\Leftrightarrow \mathcal{A} \wedge [\mathcal{B} \wedge \mathcal{C}] \\ [\mathcal{A} \vee \mathcal{B}] \vee \mathcal{C} &\Leftrightarrow \mathcal{A} \vee [\mathcal{B} \vee \mathcal{C}] \end{aligned} \qquad \text{(Assoziativgesetze)}$$

$$\begin{aligned} [\mathcal{A} \wedge \mathcal{B}] \vee \mathcal{C} &\Leftrightarrow [\mathcal{A} \vee \mathcal{C}] \wedge [\mathcal{B} \vee \mathcal{C}] \\ [\mathcal{A} \vee \mathcal{B}] \wedge \mathcal{C} &\Leftrightarrow [\mathcal{A} \wedge \mathcal{C}] \vee [\mathcal{B} \wedge \mathcal{C}] \end{aligned} \qquad \text{(Distributivgesetze)}$$

Bezüglich des Negierens gelten die folgenden Regeln.

Lemma 1.1.6. Es seien $\mathcal{A}, \mathcal{B}$ mathematische Aussagen. Dann gelten

i. $\neg(\mathcal{A} \vee \mathcal{B}) = (\neg\mathcal{A}) \wedge (\neg\mathcal{B})$
ii. $\neg(\mathcal{A} \wedge \mathcal{B}) = (\neg\mathcal{A}) \vee (\neg\mathcal{B})$

Bemerkung 1.1.7.

Negiert man eine Aussage, die mit einem All- oder Existenzquantor beginnen, so taucht in der negierten Aussage stets der andere Quantor auf. Dies ist wichtig für den Beweis von Aussagen durch Widerspruch, hier wird in der einleitenden Widerspruchsannahme die Originalaussage negiert.

1.2. Mengen

Wir beginnen mit dem Begriff der Menge, welche an dieser Stelle aufgrund einiger Komplikationen in den Details nicht im strengen mathematischen Sinne sauber definiert werden kann. Für unsere Zwecke bedienen wir uns der (naiven) Mengendefinition von Georg Cantor (1845-1918), dem Begründer der Mengentheorie:

> *Eine Menge ist eine Zusammenfassung bestimmter wohlunterschiedener Objekte unserer Anschauung oder unseres Denkens, welche Elemente genannt werden, zu einem Ganzen.*

Diese sehr einleuchtende und der alltäglichen Verwendung des Begriffs der Menge sehr nahe Umschreibung führt allerdings bei näherer Untersuchung zu Komplikationen.

Exkurs 1.2.1 (Die Russelsche Antinomie)**.**

Definiert man Mengen als "Zusammenfassung unterscheidbarer Objekte" so ergibt sich das folgende Paradoxon:

Die Menge der Mengen, welche sich nicht selbst enthalten.

Gäbe es diese Menge, und nennen wir sie A, so stellt sich die Frage:

Enthät A sich selbst?

- Beantworten wir Frage (1.2.1) mit JA so ergibt sich ein Widerspruch:
  - Wir nehmen an A enthält die Menge A (weil wir die Frage (1.2.1) mit JA beantworten).
  - Damit ist A (als Menge) keine jener erlesenen Mengen, die wir unter dem Titel "Mengen die sich nicht selbst enthalten" in A versammelt haben.
  - D.h. A ist nicht dabei, ergo: A ist (doch) nicht in A enthalten.
  - Ein Widerspruch!
- Beantworten wir Frage (1.2.1) mit NEIN so ergibt sich ein Widerspruch:
  - Wir nehmen an A enthält die Menge A nicht (weil wir die Frage (1.2.1) mit NEIN beantworten).
  - Damit ist A (als Menge) eine jener erlesenen Mengen, die wir unter dem Titel "Mengen die sich nicht selbst enthalten" in A versammelt haben.
  - D.h. A ist dabei, ergo: A ist doch in A enthalten.
  - Ein Widerspruch!

Wir beginnen mit Konventionen und Definitionen bezüglich der Notation grundlegender Begriffe im Kontext von Mengen. Seien A und B Mengen.

Definition 1.2.2 (Mengendefinitionen und -notationen)**.**

- Mengen werden mit "$\{$" und "$\}$" den *Mengenklammern* geschrieben.
- Die Schreibweise $x \in A$ bedeutet, dass x ein ***Element*** der Menge A ist.
- Ferner bedeutet $A \subset B$ (bzw. $B \supset A$), dass A eine ***Teilmenge*** von B ist, d.h. jedes Element von A ist auch ein Element von B.
- Wir nennen die Mengen A und B ***gleich***, wenn sie die gleichen Elemente enthalten.
- Eine Teilmenge A von B heißt ***echt***, wenn A nicht gleich B ist.
- Mit $A \cup B$ bezeichnen wir die ***Vereinigung*** von A und B; die Menge aller Element, die in A oder in B enthalten sind.
- Außerdem ist $A \cap B$ der ***Durchschnitt*** von von A und B; die Menge aller Elemente, die in A und B enthalten sind.
- Mit $A \setminus B$, gesprochen **"A ohne B"**, bezeichnen wir die Menge aller Elemente von A, die nicht Element von B sind (auch ***Differenz*** genannt).
- Schließlich ist $A \times B$ die ***Produktmenge*** von A und B, d.h. die Menge aller geordneten Paare (x, y) mit $x \in A$ und $y \in B$ (auch *kartesisches Produkt* genannt). Siehe auch Beispiel 1.2.3.
- Eine Menge A heißt ***endlich***, wenn A nur endlich viele Elemente besitzt.
- Die Anzahl der Elemente einer endlichen Menge A wird als die ***Kardinalität*** von A bezeichnet, und mit $|A|$

notiert (auch ***Mächtigkeit*** genannt). Ist A nicht endlich so schreibt man $|A| = \infty$.

- Die ***leere Menge*** notiert mit $\emptyset$ ist diejenige Menge, die keine Elemente enthält.
- Für eine Menge A is die ***Potenzmenge*** $\mathcal{P}(A)$ die Menge aller Teilmengen von A inklusive der leeren Menge $\emptyset$. Siehe auch Beispiel 1.2.4.
- Eine Menge ist definiert, wenn angegeben ist, welche Elemente in ihr enthalten sind. Dies kann *deskriptiv* - durch Angabe einer definierenden Eigenschaft ($A := \{n \in \mathbb{N} : n \text{ ist gerade}\}$) - und *konstruktiv* - durch Aufzählung aller in ihr enthaltenen Elemente ($B := \{2, 4, 6, 8, 10\}$) - geschehen. Wenn bei Mengen mit unendlich vielen Elementen das Bildungsgesetz klar ist, können auch unendliche Aufzählungen verwendet werden ($A := \{2, 4, 6, 8, \ldots\}$).
- Es bezeichnet $\mathbb{N} = \{1, 2, 3, \ldots\}$ die Menge der ***natürlichen Zahlen***. Es bezeichnet $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ die Menge der natürlichen Zahlen mit der Null.
- Es bezeichnet $\mathbb{Z} = \{0, -1, 1, -2, 2, -3, 3, \ldots\}$ die Menge der ***ganzen Zahlen***.
- Es bezeichnet $\mathbb{Q}$ die Menge der ***rationalen*** und $\mathbb{R}$ die Menge der ***reellen Zahlen***.

Beispiel 1.2.3 (Produktmenge)**.**

Für $A = \{1, 2, 3\}$ und $B = \{3, 4\}$ ist

$$A \times B = \{(1, 3), (1, 4), (2, 3), (2, 4), (3, 3), (3, 4)\}.$$

Beispiel 1.2.4 (Potenzmenge)**.**

Für $A = \{1, 2, 3\}$ ist

$$\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$$

In einem gewissen Sinne lässt sich mit Mengen und den ***Operatoren*** Durchschnitt und Vereinigung (Differenz und dem kartesischen Produkt) rechnen. Es gelten die folgenden "Rechenregeln".

Lemma 1.2.5. Für beliebige Mengen A, B und C gilt:

$$\left.\begin{aligned} A \cap B &= B \cap A \\ A \cup B &= B \cup A \end{aligned}\right\} \quad \text{(Kommutativgesetze)}$$

$$\left.\begin{aligned} (A \cap B) \cap C &= A \cap (B \cap C) \\ (A \cup B) \cup C &= A \cup (B \cup C) \end{aligned}\right\} \quad \text{(Assoziativgesetze)}$$

$$(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$$
$$(A \cup B) \cap C = (A \cap C) \cup (B \cap C) \qquad \text{(Distributivgesetze)}$$

Im Folgenden Beweis stehen die Symbole "$\subset$" und "$\supset$" für folgende Textüberschriften:

" $\subset$ " entspricht: "Wir Zeigen nun: linke Menge ist enthalten in rechter Menge".

" $\supset$ " entspricht: "Wir Zeigen nun: rechte Menge ist enthalten in linker Menge".

Diese Symbole sind also Abkürzungen und nicht als mathematische Symbole zu deuten.

Beweis. [Beweis von Lemma 1.2.5] Wir beweisen exemplarisch die erste der beiden in Lemma 1.2.5 als Distributivgesetz bezeichneten Gleichungen. Zu zeigen ist:

$$(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$$

Nach der Definition von Mengengleichheit müssen wir zeigen, dass die rechte menge in der linken un die linke in der rechten Menge enthalten ist.

"$\subset$": Es sei $x \in (A \cap B) \cup C$ beliebig gewählt. Für x gilt dann:

$$\begin{aligned} & x \in (A \cap B) \cup C \\ \Leftrightarrow \quad & [x \in (A \cap B) \vee x \in C] \\ \Leftrightarrow \quad & [(x \in A \wedge x \in B) \vee x \in C] \end{aligned} \tag{1.1}$$

Es gibt nun zwei Fälle:

Fall 1 $x \in C$**:** Es gilt demnach $x \in A \cup C$ und $x \in B \cup C$.
Fall 2 $x \notin C$**:** Es folgen aus (1.1) demnach sofort $x \in A$ und $x \in B$. Damit gilt auch $x \in A \cup C$ und $x \in B \cup C$.

In beiden Fällen gilt also $x \in A \cup C$ und $x \in B \cup C$ und damit $x \in (A \cup C) \cap (B \cup C)$.

Da x beliebig gewählt war gilt also allgemein für alle $x \in (A \cap B) \cup C$, dass

$$x \in (A \cap B) \cup C \qquad \Longrightarrow \qquad x \in (A \cup C) \cap (B \cup C).$$

Damit gilt nach der Defintiont von "$\subset$" also $(A \cap B) \cup C \subset (A \cup C) \cap (B \cup C)$.

"$\supset$": Es sei $x \in (A \cup C) \cap (B \cup C)$ beliebig gewählt. Für x gilt dann:

$$\begin{aligned} & x \in (A \cup C) && \cap && (B \cup C) \\ \Leftrightarrow \quad & [x \in (A \cup C) && \wedge && x \in (B \cup C)] \\ \Leftrightarrow \quad & [(x \in A \vee x \in C) && \wedge && (x \in B \vee x \in C)] \end{aligned} \tag{1.2}$$

Es gibt nun zwei Fälle:

Fall 1 $x \in C$**:** Es folgt demnach $x \in (A \cap B) \cup C$.
Fall 2 $x \notin C$**:** Es folgen aus (1.2) demnach sofort $x \in A$ und $x \in B$ und damit $x \in (A \cup C) \cap (B \cup C)$.

In beiden Fällen gilt also $x \in (A \cap B) \cup C$

Da x beliebig gewählt war, gilt also allgemein für alle $x \in (A \cup C) \cap (B \cup C)$, dass

$$x \in (A \cap B) \cup C \qquad \Leftarrow \qquad x \in (A \cup C) \cap (B \cup C).$$

Damit gilt nach der Definition von Teilmengen also $(A \cap B) \cup C \supset (A \cup C) \cap (B \cup C)$. □

Lemma 1.2.6. Sei A eine endlichen Menge mit Kardinalität n. Die Potenzmenge $\mathcal{P}(A)$ von A hat Kardinalität 2^n.

In der Mathematik lassen sich Aussagen häufig auf unterschiedliche Weisen zeigen. Für den Satz des Pythagoras sind beispielsweise mehrere hundert verschiedene Beweise bekannt. Der Satz des Pythagoras ist damit übrigens der meistbewiesene mathematische Satz. Für Lemma 1.2.6 ist ebenfalls mehr als ein Beweis bekannt. Zum einen kann man die Menge aller Teilmengen, also die Potenzmenge, mit einer zweiten Menge in Eins-zu-eins-Relation bringt. Der Trick besteht darin, dass dabei jedem Element aus der Potenzmenge genau ein Element aus der zweiten Menge und tatsächlich jedem Element aus der zweiten Menge auch ein Element der Potenzmenge zugeordnet wird. Folglich enthalten beide Mengen also genau gleich viele Elemente. Die zweite Menge stellt sich dann schlussendlich als leicht zu zählen heraus.

Neben diesem Beweis, der die im folgenden Abschnitt erinnerten Begriff im Zusammenhang von Abbildungen verwendet, existiert ein weiterer Beweis über eine Beweistechnik mit dem Namen ***vollständige Induktion***. Wir betrachten beide Beweise im Anschluss an Abschnitt 1.3.

1.3. Abbildungen

Wir starten mit grundlegenden Definitionen von Abbildungen.

Definition 1.3.1 (Abbildungen)**.** Eine ***Abbildung*** (oder auch ***Funktion***) $f : D \to B, x \mapsto f(x)$ bildet Werte aus dem ***Definitionsbereich*** D in den ***Bildbereich*** B ab. Jedem Element $x \in D$ wird durch f genau ein ***Bild*** $f(x) \in B$ zugeordnet. Gilt $f(x) = y$ für ein $y \in B$, so nennt man x das ***Urbild*** von y. Jedes $x \in D$ besitzt ein Bild $f(x)$, aber nicht jedes Element $y \in B$ muss ein Urbild besitzen.

Dies verallgemeinert man für eine Abbildung $f : D \to B$ und eine Teilmenge $Z \subset D$ ist

$$f(Z) = \{f(z) : z \in Z\}$$

die ***Bildmenge*** (manchmal auch einfach das Bild) von Z unter f. Umgekehrt bezeichnen wir für $C \subset B$ mit

$$f^{-1}(C) = \{x \in D : f(x) \in C\}$$

die ***Urbildmenge*** (manchmal auch einfach das Urbild) von C.

Abbildungen können drei ganz grundlegende Eigenschaften besitzen.

Definition 1.3.2 (Injektivität, Surjektivität, Bijektivität)**.** Eine Abbildung heißt

- ***injektiv***, wenn je zwei verschiedene $x, x' \in D$ auch verschiedene Bilder besitzen, d.h. wenn gilt:

$$x \neq x' \implies f(x) \neq f(x')$$

- ***surjektiv***, wenn jeder Bildpunkt $y \in B$ tatsächlich auch ein Urbild $x \in D$ besitzt mit $y = f(x)$, d.h. wenn gilt:

$$\forall\, y \in B\ \exists\, x \in D : f(x) = y$$

- ***bijektiv***, wenn f injektiv und surjektiv ist.

Bemerkung 1.3.3.

Injektiv: Die Definition für "injektiv" kann man sich anschaulich vorstellen als: Die Definitionsmenge D wird in die Bildmenge "injiziert", d.h. man findet für jedes $x \in D$ einen eigenen Funktionswert $y \in B$ vor, "der einmal x war".

Surjektiv: Eine surjektive Abbildung dagegen "deckt die ganze Bildmenge ab", jeder Bildpunkt wird bei einer surjektiven Abbildung auch tatsächlich angenommen. Dies ist nicht selbstverständlich: Das Wort "Bildmenge" klingt zwar wie "die Sammlung aller Bilder f (x)". Tatsächlich kann die Bildmenge aber auch einfach nur eine "grobe Schätzung" sein, wie im Beispiel $g : \mathbb{R} \to \mathbb{R}$ mit $g(x) := x^2$. Dabei sind die Werte von g stets nicht-negativ, d.h. man "erreicht" mit g nicht die ganze Bildmenge $\mathbb{R}$.

Bijektiv: Eine bijektive Abbildung stellt eine Eins-zu-Eins-Relation zwischen Definitionsmenge D und Bildmenge B her. D.h. jedes $x \in D$ hat zum einen "sein eigenes(!)" Bild $f(x) \in B$, aber auch jeder Punkt $y' \in B$ hat genau(!) ein Urbild $x' \in D$. Daraus folgt: Sind D und B endliche Mengen und sind sie über eine bijektive Abbildung $f : D \to B$ verknüpft, so haben D und B gleich viele Elemente. Der Begriff einer bijektiven Abbildung ist in der Mathematik also beim Zählen von Dingen von zentraler Bedeutung.

Es folgt eine weitere sehr umfangreiche Definition.

Definition 1.3.4. Falls f eine bijektive Abbildung ist, so hat für jedes $y \in B$ die Menge $f^{-1}(\{y\})$ genau ein Element $x \in D$ und wir schreiben einfach $x = f^{-1}(y)$. Die Abbildung $f^{-1} : B \to D, y \mapsto f^{-1}(y)$ ist in diesem Fall ebenfalls bijektiv und heißt die ***Umkehrabbildung*** von f.

Für eine Menge B und eine Zahl $k \in \mathbb{N}$ bezeichnen wir mit B^k die Menge aller Abbildungen $f : \{1, \ldots, k\} \to B$. Anstelle der Notation $f : D \to B, x \mapsto f(x)$ schreiben wir mitunter etwas lax $(f(x))_{x \in D}$. Diese Notation wird häufig verwendet, wenn $D = \{1, 2, 3, \ldots, k\}$ für eine Zahl $k \in \mathbb{N}$. Insbesondere schreiben wir die Elemente f der Menge B^k als $(f(1), \ldots, f(k))$; sie werden auch k-Tupel (und im Fall $k = 2$ Paare und im Fall $k = 3$ Tripel) genannt. Allgemeiner bezeichnen wir mit B^D die Menge aller Abbildungen $f : D \to B$. Ist $(A_i)_{i \in I}$ eine

Abbildung, die Elementen einer Menge I Teilmengen A_i einer Menge A zuordnet, so bezeichnet

$$\bigcup_{i\in I} A_i = \{x \in A : \text{ es gibt ein } i \in I \text{ mit } x \in A_i\}$$

die Vereinigung aller Mengen A_i. Analog ist

$$\bigcap_{i\in I} A_i = \{x \in A : \text{ für alle } i \in I \text{ gilt } x \in A_i\}$$

der Durchschnitt aller A_i.

Sei $f : A \to \mathbb{R}$ eine Abbildung von einer endlichen Menge $A \neq \emptyset$ in die reellen Zahlen. Dann existiert eine Bijektion $g : \{1, \ldots, k\} \to A$, wobei $k \in \mathbb{N}$. Wir definieren die ***Summe***

$$\sum_{a\in A} f(a) = f(g(1)) + f(g(2)) + \ldots + f(g(k))$$

und das ***Produkt***

$$\prod_{a\in A} f(a) = f(g(1)) \cdot f(g(2)) \cdots f(g(k)).$$

Falls A die leere Menge ist, interpretieren wir die Summe als 0 und das Produkt als 1.

Wir benötigen die Beweismethode der ***Induktion***. Die Grundlage des Induktionsprinzip ist folgende Tatsache.

Jede nicht-leere Menge natürlicher Zahlen enthält eine kleinste Zahl.

Aus dieser Tatsache folgt

Lemma 1.3.5 ("Induktionsprinzip")**.** Angenommen eine Menge $A \subset \mathbb{N}$ hat die beiden folgenden Eigenschaften.

i. $1 \in A$
ii. Wenn $1, \ldots, n \in A$, dann gilt auch $n + 1 \in A$.

Dann gilt $A = \mathbb{N}$.

Beweis. Angenommen $A \neq \mathbb{N}$. Dann ist die Menge $B = \mathbb{N} \setminus A$ nicht leer. Folglich gibt es eine kleinste Zahl $x \in B$. Aufgrund von i. ist $x \neq 1$. Ferner gilt $1, \ldots, x - 1 \in A$, weil x ja die kleinste Zahl in B ist. Nach ii. gilt also $x \in A$, im Widerspruch zu unserer Annahme, dass $x \in B$. □

Das Induktionsprinzip ermöglicht es uns, Beweise nach folgendem Schema zu führen.

i. Zeige, dass die Behauptung für $n = 1$ stimmt.
ii. Weise ferner nach, dass die Behauptung für $n + 1$ gilt, wenn sie für $1, \ldots, n$ gilt.

Dann folgt die Behauptung *für alle* $n \in \mathbb{N}$. Als Beispiel zeigen wir die *gaußsche Summenformel* (auch *kleiner Gauß* genannt).

Lemma 1.3.6 ("Kleiner Gauß"). Die Summe der ersten n natürlichen Zahlen ist

$$1 + 2 + \ldots + n = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

Beweis. Wir führen die Induktion über n.

Induktionsverankerung: Im Fall $n = 1$ ist die rechte Seite

$$\frac{1(1+1)}{2} = 1$$

was tatsächlich der Summe der ersten 1 vielen natürlichen Zahlen entspricht.

Induktionsannahme: Wir nehmen als Induktionsvoraussetzung nun an, dass die Formel für $n \in \mathbb{N}$ gilt, also dass

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}. \tag{1.3}$$

Induktionsschluss: Für den Induktionsschluss berechnen wir nun die Summe der ersten $n+1$ vielen natürlichen Zahlen

$$\begin{aligned}\sum_{i=1}^{n+1} i &= (n+1) + \sum_{i=1}^{n} i \\ &= (n+1) + \frac{n(n+1)}{2} \qquad \text{[nach Induktionsannahme (1.3)]} \\ &= \frac{2n + 2 + n^2 + n}{2} = \frac{(n+1)((n+1)+1)}{2}\end{aligned}$$

wie behauptet. □

1.3.1. Beweis von Lemma 1.2.6

Wie schon angekündigt beweisen wir Lemma 1.2.6 auf zwei unterschiedliche Wege.

Beweis. [von Lemma 1.2.6 - Variante 1] Sei A eine Menge mit n Elementen. Sei $\mathcal{W}_n$ die Menge aller "Worte" bestehend aus den Buchstaben *i* und *d* mit genau n Buchstaben.

Es sei f eine bijektive Abbildung der Elemente in A in die Menge der natürlichen Zahlen 1 bis n. Diese Abbildung ordnet die Elemente in A. Für jedes $a \in A$ existiert also eine eindeutiges $1 \leq j \leq n$ sodass $f(a) = j$.

Sei g eine Abbildung die jedem $B \in \mathcal{P}(A)$ das Wort $w \in \mathcal{W}_n$ zuordnet, sodass für alle $a \in A$ gilt

- der $f(a)$-te Buchstabe von w ist i, wenn $a \in B$ und
- der $f(a)$-te Buchstabe von w ist d, wenn $a \notin B$.

Diese Abbildung ist bijektiv.

- **Surjektivität**
  Zu zeigen: Für jedes Wort $w \in \mathcal{W}_n$ lässt sich eine Menge $B \in \mathcal{P}(A)$ finden, sodass $g(B) = w$.
  $\forall\, w \in \mathcal{W}_n\ \exists\, B \in \mathcal{P}(A) : g(B) = w.$
  Sei Q die Menge der Positionen von w, an welchen ein i steht. Sei $B = f^{-1}(Q)$. Es ist nun der $f(a)$-te Buchstabe von w ein i, wenn $a \in B$ und ein d, wenn $a \notin B$. Demnach wird B von g auf w abgebildet.
- **Injektivität**
  Zu zeigen: Es gibt keine zwei verschiedenen Mengen $B, B' \in \mathcal{P}(A)$ mit $g(B) = g(B')$.
  $\nexists B, B' \in \mathcal{P}(A) : [B \neq B' \wedge g(B) = g(B')].$
  Angenommen, es gibt zwei verschiedene Mengen $B, B' \in \mathcal{P}(A)$ sodass $g(B) = g(B')$. Dann gibt es ***ohne Beschränkung der Allgemeinheit*** ein Element $a \in B$, dass nicht in B' enthalten ist (sonst wäre B eine Teilmenge von B' - aber beide Mengen sind verschieden und somit gäbe es dann ein Element $a \in B'$, dass nicht in B enthalten ist - Umbenennung der Mengen liefert die Behauptung). Dann ist aber der $f(a)$-te Buchstabe von $g(B)$ ein i und der $f(a)$-te Buchstabe von $g(B')$ ein d und somit ist $g(B) \neq g(B')$ - ein Widerspruch zu unserer Annahme.

Wie wir schon beobachteten, haben zwei endliche Mengen genau dann die gleiche Kardinalität, wenn es eine bijektive Abbildung zwischen ihnen gibt.

Wir müssen also nur noch zählen, wie viele Wörter bestehend aus zwei Buchstaben und von der Länge n es gibt. Für jede Position gibt es 2 Möglichkeiten: Entweder steht dort ein i oder ein d. Insgesamt gibt es also 2^n unterschiedliche Wörter und somit hat die Potenzmenge einer endlichen Menge mit Kardinalität n selbst Kardinalität 2^n. □

Beweis. [von Lemma 1.2.6 - Variante 2] Wir führen die Induktion über n.

Induktionsverankerung: Im Fall $n = 1$ ist die Aussage einfach zu Prüfen. Die Potenzmenge besteht in diesem Fall aus den beiden Mengen $\emptyset$ und A selbst.

Induktionsannahme: Wir nehmen als Induktionsvoraussetzung an, dass die Potenzmenge einer Menge mit n Elementen Kardinalität 2^n habe.

Induktionsschluss: Für den Induktionsschluss nehmen wir an A habe $n + 1$ Elemente. Nun zeichnen wir ein Element $a \in A$ aus und betrachten die Menge $A' = A \setminus \{a\}$. Es gilt $|A'| = n$. Nach Induktionsvoraussetzung ist $|\mathcal{P}(A')| = 2^n$.

Wir beobachten, dass jede Teilmenge B von A entweder eine Teilmenge von A' ist oder das Element a enthält (in diesem Fall ist aber $B \setminus \{a\}$ eine Teilmenge von A'). Also können wir jeder Teilmenge $B \subset A$ genau eine Teilmenge von A' zuordnen, nämlich $B \setminus \{a\}$. Dabei wird jede Teilmenge von A' genau zwei Teilmengen von A zugeordnet. Es gibt also zweimal soviele Mengen in $\mathcal{P}(A)$ als in $\mathcal{P}(A')$. Demnach ist

$$|\mathcal{P}(A)| = |\mathcal{P}(A')| \cdot 2 = 2^n \cdot 2 = 2^{n+1}.$$

□

Bemerkung 1.3.7.

Mit der Formulierung ***ohne Beschränkung der Allgemeinheit (o. B. d. A.)*** wird zum Ausdruck gebracht, dass eine Einschränkung (z. B. des Wertebereichs einer Variablen) nur zur Vereinfachung der Beweisführung vorausgesetzt wird (insbesondere zur Verringerung der Schreibarbeit), ohne dass die Gültigkeit der im Anschluss getroffenen Aussagen in Bezug auf die Allgemeinheit darunter leidet. Der Beweis wird nur für einen von meh-

reren möglichen Fällen geführt. Dies geschieht unter der Bedingung, dass die anderen Fälle in analoger Weise bewiesen werden können (z. B. bei Symmetrie). Durch o. B. d. A. können auch triviale Sonderfälle ausgeschlossen werden.

Abschließend noch ein Wort zu Beweistechniken. Mathematische Aussagen zu beweisen erfordert neben Fleiß und Sorgfalt in der Darstellung ein hohes Maß an Kreativität. In der Beschäftigung mit mathematischen Beweisen wird man feststellen, dass es allerdings Prinzipien gibt, die immer wieder angewendet werden. Wir haben schon das Induktionsprinzip kennen gelernt. An dieser Stelle noch der Hinweis auf zwei weitere Beweisprinzipien, die zunächst ähnlich aussehen, aber zu unterscheiden sind und schon unbemerkt in obigen Beweisen auftauchten.

► **Beweis durch Kontraposition** Das logische Prinzip hinter diesem Beweisprinzip haben wir schon in Bemerkung 1.1.3 beobachtet. Um zu zeigen, dass $\mathcal{A} \Rightarrow \mathcal{B}$ kann man auch zeigen, dass $\neg\mathcal{B} \Rightarrow \neg\mathcal{A}$.

Beispiel 1.3.8.

Sei n eine gerade Quadratzahl, dann ist $\sqrt{n}$ ebenfalls gerade.

Beweis. Wir möchten zeigen:

$$\text{“}n \text{ ist gerade Quadratzahl”} \Rightarrow \text{“}\sqrt{n} \text{ ist gerade”}.$$

Die Kontraposition ist

$$\neg(\text{“}\sqrt{n} \text{ ist gerade”}) \Rightarrow \neg(\text{“}n \text{ ist gerade Quadratzahl” })$$

also

$$\text{“}\sqrt{n} \text{ ist ungerade”} \Rightarrow \text{“}n \text{ ist ungerade”}$$

Sei also $\sqrt{n}$ eine ungerade, natürliche Zahl, also gibt es ein $k_1 \in \mathbb{N}$ sodass $\sqrt{n} = 2 \cdot k_1 + 1$. Dann ist

$$\begin{aligned} n &= (\sqrt{n})^2 \\ &= (2k_1 + 1)^2 \\ &= 4k_1^2 + 4k + 1 \\ &= 2(2k_1^2 + 2k_1) + 1 \end{aligned}$$

Setzte $k_2 = 2k_1^2 + 2k_1$. Dann ist $n = 2k_2 + 1$ also ungerade, was zu zeigen war. □

► **Beweis durch Widerspruch** Dabei möchte man die Wahrheit einer Aussage $\mathcal{A}$ beweisen und nimmt die Negation von $\mathcal{A}$, nämlich $\neg\mathcal{A}$ als wahr an und führt dies über logische Schlüsse zu einem Widerspruch. Also kann $\neg\mathcal{A}$ nicht wahr sein, also muss $\mathcal{A}$ wahr sein.

Beispiel 1.3.9.

Ein Beispiel findet man in der Variante 1 des Beweises von Lemma 1.2.6: Nachweis der Injektivität von g.

1.4. Relationen

Objekte, die zu unterscheiden sind, können dennoch in Bezug zueinander stehen. Der mathematische Begriff der Relation misst dieses „in (einem bestimmten) Bezug zueinander stehen“, er gibt für zwei Objekte entweder die Antwort „Ja, die beiden Objekte stehen in (in diesem bestimmten) Bezug zueinander“ oder „Nein, die Objekte stehen nicht (in diesem bestimmten) Bezug zueinander“. Wir erinnern zunächst an den Begriff des Kartesisches Produkts. Für zwei Mengen A, B heißt $A \times B := \{(a, b) \ : \ a \in A \wedge b \in B\}$ das **kartesische Produkt** von A und B.

Definition 1.4.1. Eine **(binäre) Relation** zwischen zwei Mengen A und B ist eine Teilmenge $R \subseteq A \times B$. Im Falle $A = B$ spricht man von **einer Relation auf** A.

Eine Relation zwischen A und B ist also eine Teilmenge aller *geordneten* Paare der Form (a, b) mit $a \in A$ und $b \in B$.

Beispiel 1.4.2.

- Der Begriff „größer als“ induziert eine Relation auf der Menge aller Zahlen $\mathbb{N}$:

$$R := \{(a, b) \in \mathbb{N} \times \mathbb{N} \ : \ a > b\} = \{(2, 1), \ (3, 1), (3, 2), \ (4, 1), (4, 2), (4, 3), \ (5, 1), \ldots\}$$

- Funktionen sind Relationen:
  Die Paare aus Wert $x \in \mathbb{R}$ und Funktionswert $f(x)$ einer Funktion $f : \mathbb{R} \to \mathbb{R}$ bilden eine Relation auf $\mathbb{R}$

$$R := \{(x, y) \in \mathbb{R} \times \mathbb{R} \ : \ f(x) = y\}$$

- Der Begriff „verwandt sein mit“ (engl.: „related to“) beschreibt eine Relation auf der Menge aller Menschen.

Definition 1.4.3. Eine Relation auf A heißt

- **reflexiv**, wenn für alle $a \in A$ gilt

$$(a, a) \in R.$$

- **symmetrisch**, wenn für alle $a, b \in A$ gilt

$$(a, b) \in R \Rightarrow (b, a) \in R.$$

- **transitiv**, wenn für alle $a, b, c \in A$ gilt

$$(a, b) \in R \text{ und } (b, c) \in R \Rightarrow (a, c) \in R.$$

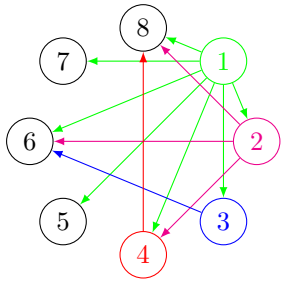
Beispiel 1.4.4.

- Wir betrachten die Teilbarkeitsrelation $R := \{(a, b) \in \mathbb{N} \times \mathbb{N} \ : \ a \text{ teilt } b\}$ auf der Menge der natürlichen Zahlen.
  Diese Relation ist reflexiv und transitiv, aber nicht symmetrisch.
- Auf $\mathbb{N}$ definiert $(a, b) \in R :\Leftrightarrow a > b$ die Relation $R := \{(a, b) \in \mathbb{N} \times \mathbb{N} \ : \ a > b\}$.
  Die Relation ist transitiv aber nicht reflexiv und nicht symmetrisch.

Bemerkung 1.4.5.

Im Fall einer endlichen Menge A kann eine Relation auf A durch einen gerichteten Graphen (mit möglichen Schlingen) visualisiert werden: Von einem Element $a \in A$ führt genau dann ein Bogen (gerichtete Kante) zu einem Element $b \in A$, wenn $(a, b) \in R$ gilt. Der gerichtete Graph in der unteren Abbildung zeigt den Graphen zur Teilbarkeitsrelation auf $\{1, \ldots 8\}$, d.h. den Graphen für

$$R := \{(a, b) \in \{1, \ldots, 8\} \times \{1, \ldots, 8\} : a \neq b \text{ und } a \text{ teilt } b\}.$$



1.4.1. Äquivalenzrelationen

Das Ziel bei der Verwendung von Äquivalenzrelationen ist, den Begriff „gleich“ (im Sinne von identisch) zu verallgemeinern auf „ähnlich“ bzw. „gleich bezüglich einer Eigenschaft“. So können z.B. zwei Gegenstände gleich sein im Bezug auf ihre Farbe (also die gleiche Farbe haben) ohne jedoch identisch zu sein.

Um den Begriff „gleich“ auf „ähnlich“ zu verallgemeinern, müssen wir sicherstellen, dass für die Verallgemeinerung weiter gilt, dass

- ein Gegenstand stets zu sich selbst ähnlich ist. (Reflexivität)
- wenn a zu b ähnlich ist, dann auch b zu a. (Symmetrie)
- wenn a ähnlich ist zu b und dies wiederum ähnlich ist zu c, so ist auch a ähnlich zu c. (Transitivität)

Definition 1.4.6 (Äquivalenzrelation)**.** Eine Relation heißt ***Äquivalenzrelation***, wenn sie reflexiv, symmetrisch und transitiv ist.

Wir schauen uns einige Beispiel von Relationen an.

Beispiel 1.4.7.

Es sei A die Menge aller Schüler einer Schule und

$$R := \{(a, b) \in A \times A \ : \ a \text{ ist in derselben Schulklasse wie } b\} \ .$$

Die Relation R ist eine Äquivalenzrelation

- Jeder Schüler $a \in A$ ist Schüler seiner (eigenen) Schulklasse, es gilt also $(a, a) \in R$. (Reflexivität)
- Ist $(a, b) \in R$, so ist Mitschüler a in derselben Schulklasse wie b.
  Also ist auch $(b, a) \in R$, denn b ist umgekehrt auch in derselben Schulklasse wie a. (Symmetrie)
- Wenn $(a, b) \in R$ und $(b, c) \in R$ gelten, dann gilt auch $(a, c) \in R$, denn es ist a in derselben Schulklasse wie b und b in derselben Schulklasse wie c, und das heißt a ist in der Schulklasse von c. (Transitivität)

Beispiel 1.4.8.

Es sei $R := \{(n, m) \in \mathbb{N} \times \mathbb{N} \ : \ m \text{ hat denselben Rest beim Teilen durch 2 wie } n\}$.

Die Relation R ist eine Äquivalenzrelation

- Jede Zahl $n \in \mathbb{N}$ hat einen festen Rest beim Teilen durch 2, es gilt also $(n, n) \in R$. (Reflexivität)
- Ist $(n, m) \in R$, so hat m denselben Rest beim Teilen durch 2 wie n.
  Also ist auch $(m, n) \in R$, denn n hat denselben Rest beim Teilen durch 2 wie m. (Symmetrie)
- Wenn $(n, k) \in R$ und $(k, m) \in R$ gelten, dann gilt beim Teilen durch 2:
  k hat denselben Rest wie n und m denselben Rest wie k, und das heißt m hat denselben Rest wie n. (Transitivität)

Die beiden „Schulklassen“ in den die Zahlen hier wahlweise gehen heißen „Gerade Zahlen“ oder „Ungerade Zahlen“.

Formal korrekt, aber trotzdem seltsam, ist das folgende Beispiel:

Beispiel 1.4.9 (Jede Relation über die Leere Menge ist eine Äquivalenzrelation)**.**

Es sei $A := \emptyset$ und $R \subseteq A \times A$. Dann ist R eine Äquivalenzrealtion.

Zunächst gilt besondererweise $R = \emptyset$ denn $A \times A = \emptyset \times \emptyset = \emptyset$.

- Reflexivität: Wegen $A = \emptyset$ gibt es *kein* $a \in A$.
  Also gibt es insbesondere kein $a \in A$ mit der Zusatzeigenschaft $(a, a) \notin R$.

Kein $a \in A$ verletzt also die Reflexivität von R. R ist also reflexiv.

- Symmetrie: Wegen $R = \emptyset$ gibt es kein Paar $(a, b) \in R$.
  Also gibt es insbesondere kein Paar $(a, b) \in R$ mit der Zusatzeigenschaft $(b, a) \notin R$.
  Kein Paar $(a, b) \in R$ verletzt also die Symmetrie von R. R ist also symmetrisch.
- Transitivität: Wegen $R = \emptyset$ gibt es kein Paar $(a, b) \in R$.
  Also gibt es insbesondere kein $(a, b) \in R$ mit der Zusatzeigenschaft, dass es $(b, c) \in R$ gibt.
  Also gibt es kein Paar (a, b) für dass zwar $(b, c) \in \mathbb{R}$ gilt, aber nicht $(a, c) \in \mathbb{R}$.
  Kein Paar $(a, b) \in R$ verletzt also die Transitivität von R. R ist also transitiv.

Beispiel 1.4.10 (keine Äquivalenzrelation)**.**

Die Relation $R := \{(n, m) \in \mathbb{N} \times \mathbb{N} \ : \ n \leq m\}$ ist **nicht symmetrisch** und deswegen *keine* Äquivalenzrelation:

Es gilt zwar $(1, 2) \in R$ wegen $1 \leq 2$ aber es gilt nicht $(2, 1) \notin R$.

Die Relation R ist zwar reflexiv und transitiv, aber eben nicht symmetrisch.

Bemerkung 1.4.11.

Am Gegenbeispiel 1.4.10 sieht man:

> *Um zu zeigen, dass eine Relation* keine *Äquivalenzrelation ist, genügt es zu zeigen, dass eine der benötigten Eigenschaften nicht gilt. Um wiederum zu zeigen, dass eine Relation z.B. nicht symmetrisch ist, genügt ein einziges Gegenbeispiel!*

Definition 1.4.12 (Äquivalenz)**.** Es sei R eine Äquivalenzrelation auf A.

Für $(a, b) \in R$ schreibt man kurz $a \sim_R b$, und sagt: a und b sind ***äquivalent bezüglich*** R.

Wenn klar ist, welche Relation Gemeint ist, wird „$\sim_R$“ zu „$\sim$“ vereinfacht.

Das Symbol $\sim$ verallgemeinert das Symbol „$=$“, es gelten auf Ebene der logischen Operatoren die selben Regeln:

$a = b$	ist äquivalent zu	$b = a$.	(Symmetrie)	Aus $a = b$ und $b = c$	folgt $a = c$.	(Transitivität)
$a \sim b$	ist äquivalent zu	$b \sim a$.		Aus $a \sim b$ und $b \sim c$	folgt $a \sim c$.	

Definition 1.4.13 (Äquivalenzklasse)**.** Es sei R eine Äquivalenzrelation auf A.

Für jedes Element $a \in A$ ist die ***Äquivalenzklasse***

$$[a]_R := \{b \in A \ : \ (a, b) \in R\}$$

die Menge der zu a äquivalenten (bzw. ähnlichen) Elemente aus A.

Beispiel 1.4.14 (Schul**klassen** sind Äquivalenz**klassen**).

Es sei A die Menge aller Schüler einer Schule und

$$R := \{(a,b) \in A \times A \ : \ a \text{ ist in derselben Schulklasse wie } b\} \ .$$

Für jeden Schüler a bilden die Mitschüler seiner Schulklasse seine Äquivalenzklasse:

$$\begin{aligned} [a]_R &= \{b \in A \ : \ (a,b) \in R\} \\ &= \{b \in A \ : \ b \text{ ist in derselben Schulklasse wie } a\} = \big\{ \text{ Alle Schüler in der Schulkasse von } a \big\} \end{aligned}$$

Gilt $(a,b) \in R$, d.h. a und b sind in derselben (Schul-)Klasse (z.B. “6c”), so gilt $[a]_R = [b]_R$:

$$\begin{aligned} [a]_R &= \big\{ \text{ Alle Schüler in der Schulklasse von } a \big\} = \big\{ \text{ Alle Schüler der 6c } \big\} \\ [b]_R &= \big\{ \text{ Alle Schüler in der Schulklasse von } b \big\} = \big\{ \text{ Alle Schüler der 6c } \big\} \end{aligned}$$

Beispiel 1.4.15.

Es sei $R := \{(n,m) \in \mathbb{N} \times \mathbb{N} \ : \ m \text{ hat denselben Rest beim Teilen durch 2 wie } n\}$.

- Die Zahl 7 hat einen Rest von 1 beim Teilen durch 2, es gilt also:

$$\begin{aligned} [7]_R &= \{m \in \mathbb{N} \ : \ (7,m) \in R\} \\ &= \{m \in \mathbb{N} \ : \ m \text{ hat selben Rest beim Teilen durch 2 wie } 7\} \\ &= \{m \in \mathbb{N} \ : \ m \text{ hat Rest 1 beim Teilen durch 2}\} \\ &= \big\{ \text{ Alle ungeraden Zahlen} \big\} \end{aligned}$$

  Analog gilt $[3]_R = \big\{ \text{ Alle ungeraden Zahlen} \big\}$ und $[5]_R = \big\{ \text{ Alle ungeraden Zahlen} \big\}$ etc.
- Die Zahl 8 hat einen Rest von 0 beim Teilen durch 2, es gilt also:

$$\begin{aligned} [8]_R &= \{m \in \mathbb{N} \ : \ (8,m) \in R\} \\ &= \{m \in \mathbb{N} \ : \ m \text{ hat selben Rest beim Teilen durch 2 wie } 8\} \\ &= \{m \in \mathbb{N} \ : \ m \text{ hat Rest 0 beim Teilen durch 2}\} \\ &= \big\{ \text{ Alle geraden Zahlen} \big\} \end{aligned}$$

  Analog gilt $[2]_R = \big\{ \text{ Alle geraden Zahlen} \big\}$ und $[6]_R = \big\{ \text{ Alle geraden Zahlen} \big\}$ etc.

Die beiden „Schulklassen",’ in die die Zahlen hier wahlweise gehen, heißen „Grade Zahlen“ oder „Ungrade Zahlen“.

Anhand der Beispiele erkennt man bereits, dass zwei Äquivalenzklassen entweder „grundverschieden“ sind (leerer Schnitt) oder aber identisch sind. Dies ist einleuchtend, denn die Äquivalenzklasse $[a]_R$ besteht aus allen Elementen, die zu a äquivalent sind, d.h. gleich sind bezüglich der Eigenschaft, die R definiert.

Fordert man z.B. „1) Nenne alles, was dieselbe Länge hat wie a.“ und erhält als Antwort unter anderem b, so bekommt man auf die Aufforderung „2) Nenne alles, was dieselbe Länge hat wie b.“ dieselbe Antwort wie bei 1). Dies kann man dann mit beliebigen Eigenschaften wiederholen (Gewicht, Eckenanzahl etc.).

Diese Einsicht verallgemeinern wir zu „Zwei Elemente aus A haben bezüglich R genau dann dieselbe Äquivalenzklasse, wenn sie bezüglich R äquivalent sind“. Genauer gilt:

Lemma 1.4.16. Es sei R eine Äquivalenzrelation über A.

i) Für $a, b \in A$ gilt dann entweder $[a]_R = [b]_R$ oder $[a]_R \cap [b]_R = \emptyset$.
ii) Es gilt $[a]_R = [b]_R$ genau dann wenn $(a, b) \in R$ gilt.

Beweis. Es sei R eine Äquivalenzrelation über A und $a, b \in R$ mit $a \neq b$.

Wir zeigen zunächst ii).

'$\Rightarrow$' Annahme: Es gelte $[a]_R = [b]_R$. Wegen $(b, b) \in R$ gilt $b \in [b]_R$ und damit $b \in [a]_R$ bzw. $(a, b) \in R$.

'$\Leftarrow$' Annahme: Es gelte $(a, b) \in R$. Wir zeigen $[a]_R \subseteq [b]_R$ und $[b]_R \subseteq [a]_R$.
Sei nun $x \in [a]_R$, so gilt $(a, x) \in R$ und da R eine Äquivalenzrelation ist gilt damit

$$(a, x), (a, b) \in R \quad \Rightarrow \quad (x, a), (a, b) \in R \quad \Rightarrow \quad (x, b) \in R \quad \Rightarrow \quad x \in [b]_R$$

Da x beliebig war folgt also $[a]_R \subseteq [b]_R$. Ganz analog beweist man $[b]_R \subseteq [a]_R$. Es gilt also: $[a]_R = [b]_R$.

Wir zeigen nun i): Es seien $a, b \in A$ mit $[a]_R \cap [b]_R \neq \emptyset$. Zu zeigen ist: $[a]_R = [b]_R$.

Sei $c \in [a]_R \cap [b]_R$ beliebig. Nach Def. der Äquivalenzklassen gilt: $(a, c), (b, c) \in R$ und da R eine Äquivalenzrelation ist gilt damit

$$(a, c), (b, c) \in R \quad \Rightarrow \quad (a, c), (c, b) \in R \quad \Rightarrow \quad (a, b) \in R$$

Aus ii) schließen wir nun: $[a]_R = [b]_R$. □

Aus Lemma 1.4.16 schließen wir, dass es genügt, ein *einziges* Element einer Äquivalenzklasse zu kennen, um die Äquivalenzklasse zu rekonstruieren:

Definition 1.4.17 (Vertreter einer Äquivalenzklasse)**.** Es sei $M \subseteq A$ eine Äquivalenzklasse einer Äquivalenzrelation R auf der Menge A.

Ein Element $x \in M$ heißt dann ***Vertreter*** der Äquivalenzklasse M, denn es gilt $[x]_R = M$.

Beispiel 1.4.18 (Jeder Schüler ist Vertreter seiner Schul**klasse**)**.**

Es sei A die Menge aller Schüler einer Schule und

$$R := \{(a, b) \in A \times A \ : \ a \text{ ist in derselben Schulklasse wie } b\} .$$

Es sei nun Alice eine Schülerin der Schulklasse “6c”. Alle Mitschüler aus der Schulklasse von Alice bilden die Äquivalenzklasse von Alice:

$$\begin{aligned}[\text{Alice}]_R &= \{b \in A \ : \ (\text{Alice}, b) \in R\} \\ &= \{b \in A \ : \ b \text{ ist in derselben Schulklasse wie Alice } \} = \{ \text{ Alle Schüler der 6c } \}\end{aligned}$$

Jeder Schüler und jede Schülerin aus der 6c ist nun ein *Vertreter* seiner (Schul-)Klasse, denn anhand des einzelnen Schülers kann man natürlich die ganze Klasse ermitteln:

Ist Bob auch in der 6c, d.h. Alice und Bob sind in derselben (Schul-)Klasse “6c”, so gilt $[\text{Alice}]_R = [\text{Bob}]_R$:

$$[\text{Bob}]_R = \{ \text{ Alle Schüler in der Schulklasse von Bob } \} = \{ \text{ Alle Schüler der 6c } \}$$

Beispiel 1.4.19.

Es sei $R := \{(n, m) \in \mathbb{N} \times \mathbb{N} \ : \ m \text{ hat denselben Rest beim Teilen durch 2 wie } n\}$.

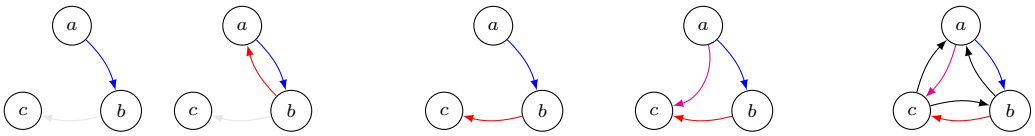
Die beiden Äquivalenzklassen von R sind:

$$\begin{aligned} M_g &:= \{2n \ : \ n \in \mathbb{N}\} \quad \text{(alle graden Zahlen)} \quad \text{und} \\ M_u &:= \{2n+1 \ : \ n \in \mathbb{N}\} \quad \text{(alle ungraden Zahlen)} \end{aligned}$$

Ein Vertreter von M_g ist $x = 6 \in M_g$, und es gilt tatsächlich $[6]_R = M_g$ (s. Beispiel 1.4.15).

1.4.2. Äquivalenzklassen: Veranschaulichung als Graph

Die Einsichten aus Lemma 1.4.16 lassen sich leichter anhand von Graphen veranschaulichen bzw. verstehen. Ist R eine Äquivalenzrealtion so „erzeugen“ die Regeln der Symmetrie und Transitivität Bögen im zugehörigen gerichteten Graphen der Relation:



$(a, b) \in R \quad \Rightarrow (b, a) \in R$ Symmetrie

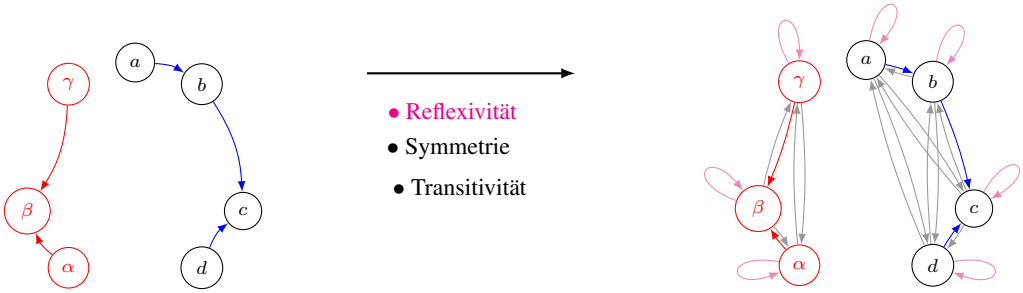
$(a, b), (b, c) \in R \quad \Rightarrow (a, c) \in R$ Transitivität

Transitivität & Symmetrie

Aus der Skizze entnimmt man: Ist G der gerichtete Graph einer Äquivalenzrelation und sind zwei Knoten v und w *irgendwie* über einen Weg aus Bögen (und Gegenbögen) verbunden, so sind v und w in$R$ auch direkt verbunden. Dies liefert:

Der gerichtetete Graph G einer Äquivalenzrelation zerfällt in disjunkte *Cliquen* G'_i:

- Jeder dieser Untergraphen G'_i ist eine Clique (bzw. vollständig), d.h. jeder mögliche Bogen der Form (a, b) mit Knoten von G'_i ist Teil des Graphen.
- Je zwei verschiedene Untergraphen G'_i und G'_j sind disjunkt: D.h. es gibt keine Bögen von G_i nach G_j (und umgekehrt).



Bekannte Paare einer Relation

Kleinste mögliche Relation mit allen vorgegebenen Paaren.

Abbildung 1.1.: Kleinstmögliche Äquivalenzrelation mit einigen bekannte Tupeln.

Jede dieser Cliquen ist eine Äquivalenzklasse der Äquivalenzrelation R. Im Falle der Relation $R = \{(a, b) : b$ spielt im selben Verein wie $a\}$ sind dies grade die möglichen Vereine.

Teil I.

Diskrete Mathematik

2 Rechnen mit ganzen Zahlen - Anwendungen

2.1. Grundlagen

In den folgenden Abschnitte werden wir Anwendungen in der Diskreten Mathematik kennenlernen, welche auf Grundlagen der Algebra, der linearen Algebra und elementaren Zahlentheorie aufbauen. Wir erinnern einige Objekte aus der Mathematik für die Informatik I.

Erinnerung

Modulo-Rechnung

Zunächst Erinnern wir an die grundlegende Definition.

Definition 2.1.1 (Rest und Äquivalenz Modulo m)**.** Es seinen $a, b \in \mathbb{Z} \setminus \{0\}$. Mit $\mathsf{Rest}(a, b)$ bezeichenen wir den ***Rest***, der beim Teilen von a durch b entsteht:

$$\mathsf{Rest}(a, b) := \min\{r \in \mathbb{N} \ : \ \exists m \in \mathbb{Z} \text{ mit } \ a = m \cdot b + r\}.$$

Es sei $m \in \mathbb{N} \setminus \{1\}$, dann gilt für $a, b \in \mathbb{Z}$

$$a \equiv b \pmod{m} \quad \Leftrightarrow \mathsf{Rest}(a, m) = \mathsf{Rest}(b, m)$$

Man sagt in diesem Fall „a und b sind äquivalent *mod* m“.

Die Zahl m bezeichnet man dabei als ***(den) Modul*** der *Modul-Gleichung* $a \equiv b \pmod{m}$.

Die Relation Äquivalenz Modulo m ist eine Äquivalenzrelation.

Lemma 2.1.2. Für jedes $m \in \mathbb{N} \setminus \{1\}$ ist $\mathcal{R}_m := \{(a, b) \in \mathbb{Z} \times \mathbb{Z} \ : \ a \equiv b \pmod{m}\}$ eine Äquivalenzrelation.

Euklidischer Algorithmus und der Satz von Bézout

Der Euklidischen Algorithmus berechnet neben dem $\mathrm{ggT}(a, b)$ zwei ganze Zahlen $s, t \in \mathbb{Z}$, für die $s \cdot a + t \cdot b = \mathrm{ggT}(a, b)$ gilt. Diese Zahlen s, t heißen ***Bézout-Multiplikatoren***. Wir werden uns bald mit dem RSA-Verfahren (ein Kryptoverfahren) und den Chinesischen Restsatz beschäftigen. Die ***Bézout-Multiplikatoren*** spielen dabei eine wichtige Rolle.

Algorithmus 2.1.3 (Erweiterter Euklidischer Algorithmus)**.**

Input: $a, b \in \mathbb{N}$ mit $a \geq b$.
Output: $r_{j-1} = \mathrm{ggT}(a, b)$ und die Gleichung $r_{j-1} = s_{j-1} \cdot a + t_{j-1} \cdot b$.
Setze $r_0 := a$ und $r_1 := b$.
Setze $s_0 := 1$ und $s_1 := 0$.
Setze $t_0 := 0$ und $t_1 := 1$.
Setze $j := 1$.
while $r_j > 0$ **do**
Setze $m_j := \lfloor \frac{r_{j-1}}{r_j} \rfloor$.
Setze $r_{j+1} := r_{j-1} - m_j r_j$.
Setze $s_{j+1} := s_{j-1} - m_j s_j$.
Setze $t_{j+1} := t_{j-1} - m_j t_j$.
Setze $j := j + 1$.
end while
return $(r_{j-1}, s_{j-1}, t_{j-1})$.

Lemma 2.1.4. Es sei (r_n, s_n, t_n) der Output des Algorithmus 2.1.3 mit Input $a, b \in \mathbb{N}$ mit $a \geq b$. Dann gilt für die Zwischenergebnisse

$$r_i = s_i \cdot a + t_i \cdot b \qquad \text{für alle } i \in \{0, 1, \ldots, n\}.$$

Insbesondere gilt also $r_n = \mathrm{ggT}(a, b) = s_n \cdot a + t_n \cdot b$.

Satz 2.1.5 (Satz von Bézout)**.** Für zwei Zahlen $a, b \in \mathbb{N}$ gibt es Zahlen $s, t \in \mathbb{Z}$ mit $s \cdot a + t \cdot b = \mathrm{ggT}(a, b)$.

Der Eukidischen Algorithmus in Form von Algorithmus 2.1.3 lässt sich wie im Anhang gezeigt ANHANG angenehm übersichtlich in Tabellenform durchführen.

2.1.1. Konsequenzen des Satzes von Bézout: Lemma von Euklid

Bevor wir uns mit dem Chinesischen Restsatz beschäftigen, ernten wir noch eine zahlentheoretische Aussage über die Faktorisierung natürlicher Zahlen.

Lemma 2.1.6 (Euklid)**.** Teilt eine Primzahl p das Produkt $a \cdot b$ zweier Zahlen $a, b \in \mathbb{N}$, so teilt p auch mindestens einen der Faktoren a oder b.

Beweis. Wir nehmen an, dass p kein Teiler von a ist und zeigen, dass dann $p|b$ gelten muss:

Es gelte $p \nmid a$. Da p Prim ist gilt dann $\mathrm{ggT}(a, p) = 1$. Nach Satz von Bézout gibt es damit $s, t \in \mathbb{Z}$ mit $s \cdot p + t \cdot a = 1$. Multiplikation mit b auf beiden Seiten der Gleichung ergibt:

$$s \cdot p \cdot b + t \cdot (\underbrace{a \cdot b}_{\text{nach Voraussetzung Vielfaches von } p}) = b$$

Da p beide beide Summanden auf der linken Seite teilt, teilt p auch die rechte Seite, also b. □

Induktiv ergibt sich daraus unmittelbar das folgende Korollar.

Korollar 2.1.7. Ist ein Produkt $\prod_{i=1}^{m} a_i$ von $m > 2$ ganzen Zahlen durch die Primzahl p teilbar, so ist mindestens ein Faktor a_i durch p teilbar.

2.2. Der Chinesische Restsatz

Mit Hilfe des nachfolgend diskutierten Chinesischen Restsatzes ist es möglich, Berechnungsprobleme mit großen Zahlen in mehrere Probleme mit kleineren Zahlen aufzuteilen:

> *Statt mit der ganzen Zahl $a \in \mathbb{Z}$ zu rechnen, rechnet man mit den Resten $(a_1, \ldots, a_k)$ bezüglich paarweise teilerfremder Moduln $m_1, \ldots, m_k$.*

Diese Strategie setzt aber voraus, dass man nach Durchführung der "vereinfachten Rechnung mit Resten", die ursprünglich zu berechnende Zahl aus den Resten "zurückgewinnen kann". Eine Antwort auf diese Frage der Rekonstruierbarkeit wird durch den ***Chinesischen Restsatz*** gegeben, der in einem speziellen Fall bereits Sun Tsu etwa 300 n. Chr. bekannt war.

Im Folgenden werden wir Systeme von Modulgleichungen der Form $x \equiv a_i \pmod{m_i}$ lösen. Solche Systeme nennt man „simultane Kongruenzen", da x gleichzeitig (also simultan) mehrere Kongruenzen erfüllen muss.

Soll ein $x \in \mathbb{Z}$ mehrere solcher Modulgleichungen mit verschiedenen m_i erfüllen, so ist dies genau dann ***nicht immer möglich***, wenn die m_i einen gemeinsamen Teiler haben, wie das folgende Beispiel zeigt.

Beispiel 2.2.1.

Das folgende System von Modulgleichungen ist nicht lösbar:

$$\begin{aligned} x &\equiv 0 \pmod 6 \qquad x \text{ ist gerade wegen } x = k \cdot 6 + 0 \\ x &\equiv 1 \pmod 8 \qquad x \text{ ist ungerade wegen } x = \ell \cdot 8 + 1 \end{aligned}$$

Die erste Gleichung impliziert, dass x gerade ist, aber x muss ungerade sein, um die zweite Gleichung zu erfüllen.

Das folgende System von Modulgleichungen ist dagegen mit $x = 10$ lösbar:

$$\begin{aligned} x &\equiv 4 \pmod 6 \\ x &\equiv 2 \pmod 8 \end{aligned}$$

Sind die entsprechenden Module jedoch teilerfremd, so garantiert der Chinesische Restsatz immer eine Lösung:

Satz 2.2.2 (Chinesischer Restsatz)**.** Seien $m_1, \ldots, m_k \in \mathbb{N}$ paarweise teilerfremd und $a_1, \ldots, a_k \in \mathbb{Z}$.

- Dann existiert eine Lösung $x = a$ für $a \in \mathbb{Z}$ des Systems von Kongruenzen

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

- Die Lösung ist modulo $m := m_1 \cdot m_2 \cdot \ldots \cdot m_k$ eindeutig, d.h. die Lösungsmenge ist $\{a + \ell \cdot m \ : \ \ell \in \mathbb{Z}\}$.

Ein typischer Fehler in der Analyse des Chinesischen Restsatzes ist, fälschlich die Umkehrung des Satzes anzunehmen.

Bemerkung 2.2.3 (Umkehrung und Berechenbarkeitsaspekt des Chinesischen Restsatzes)**.**

Es gilt

NICHT paarweise teilerfremde Moduln. $\implies$ *System* **kann** *lösbar sein,* **muss aber nicht**.

Hat ein System von Kongruenzen ***nicht*** paarweise teilerfremde Moduln $m_1, \ldots, m_k$, so kann das System trotzdem lösbar sein (s. Beispiel 2.2.1), wenn die Reste $a_1, \ldots, a_k$ geeignet gewählt wurden.

Betrachtet man den Chineschen Restsatz unter dem Aspekt "praktischer Berechenbarkeit" so übersieht man leicht die wichtige Implikation der zweiten Aussage.

Es gilt

DOCH paarweise teilerfremde Moduln. $\implies$ *Lösungen liegen "sehr weit" auseinander!*

Hat ein System von Kongruenzen paarweise teilerfremde Moduln $m_1, \ldots, m_k$, so liegen je zwei Lösungen des Systems "*sehr, sehr weit*" von einander entfernt. Die Lösungen unterscheiden sich nämlich um ein Vielfaches von $m := m_1 \cdot m_2 \cdot \ldots \cdot m_k$, die Lösungen liegen also mindestens "m-weit" auseinander.

Wir beweisen nun den Chinesischen Restsatz.

Beweis. Gegeben sie das System simultaner Kongruenzen aus Satz 2.2.2 mit $m_1, \ldots, m_k \in \mathbb{N}$ paarweise teilerfremd und $a_1, \ldots, a_k \in \mathbb{Z}$.

- **Eindeutigkeit:** Sind x und $\widetilde{x}$ Lösungen von $(\star)$, d.h. $x \equiv \widetilde{x} \equiv a_i \pmod{m_i}$ für alle i, so gilt $m_i | (x - \widetilde{x})$ für alle i. Aus der Eindeutigkeit der Primfaktorzerlegung und der paarweisen Teilerfremdheit der m_i folgt $m | (x - \widetilde{x})$, d.h. $x - \widetilde{x} \equiv 0 \pmod{m}$.
- **Existenz:** Der nachfolgende Existenzbeweis liefert gleichzeitig ein Verfahren zur Bestimmung von x. Wir bestimmen im folgenden Basislösungen e_ℓ, $1 \leq \ell \leq k$, dies sind Lösungen des speziellen Glei-

chungssystems $e_\ell = 1 \pmod{m_\ell}$ und $e_j = 0 \pmod{m_j}$ für $j \neq \ell$. Es gelten dann:

e_1	e_2	$\dots$	e_k
$e_1 \equiv 1 \pmod{m_1}$	$e_2 \equiv 0 \pmod{m_1}$		$e_k \equiv 0 \pmod{m_1}$
$e_1 \equiv 0 \pmod{m_2}$	$e_2 \equiv 1 \pmod{m_2}$		$e_k \equiv 0 \pmod{m_2}$
$\vdots$	$\vdots$	$\dots$	$\vdots$
$e_1 \equiv 0 \pmod{m_k}$	$e_2 \equiv 0 \pmod{m_k}$		$e_k \equiv 1 \pmod{m_k}$

Aus dieser setzen wir die Lösung wie folgt zusammen:

$$x := a_1 \cdot e_1 + a_2 \cdot e_2 + \ldots + a_k \cdot e_k$$

Es gilt dann:

$$\begin{array}{lllll}
x \equiv a_1 \cdot 1 & +a_2 \cdot 0 & +\ldots+ & a_k \cdot 0 & \pmod{m_1} \quad x \equiv a_1 \pmod{m_1} \\
x \equiv a_1 \cdot 0 & +a_2 \cdot 1 & +\ldots+ & a_k \cdot 0 & \pmod{m_2} \quad x \equiv a_2 \pmod{m_1} \\
\vdots & & \ddots & & \qquad\qquad \vdots \\
x \equiv a_1 \cdot 0 & +a_2 \cdot 0 & +\ldots+ & a_k \cdot 1 & \pmod{m_2} \quad x \equiv a_k \pmod{m_k}
\end{array}$$

Berechnung der Basislösungen: Es sei $\ell \in \{1, \dots k\}$. Setze

$$M_\ell := \prod_{\substack{i=1 \\ i \neq \ell}}^{k} m_i = \frac{m_1 \cdot m_2 \cdot \ldots \cdot m_k}{m_\ell} \qquad \text{(Produkt aller } m_i \textbf{ außer } m_\ell\text{)}$$

Da die Moduln $m_1, m_2, \dots, m_k$ paarweise teilerfremd sind, gilt $\mathrm{ggT}(m_\ell, M_\ell) = 1$. Nach dem Satz von Bézout existieren daher ganze Zahlen s, t, so dass gilt:

$$1 = s \cdot m_\ell + \underbrace{t \cdot M_\ell}_{e_\ell}$$

Diese Zahlen s, t können mit dem erweiterten Euklidischen Algorithmus berechnet werden. Wir setzen nun $e_\ell := t \cdot M_\ell = 1 - s \cdot m_\ell$. Damit gilt:

$$\begin{array}{llllll}
e_\ell & = & 1 - s \cdot m_\ell & \equiv & 1 & \pmod{m_\ell} \\
e_\ell & = & t \cdot M_\ell & \equiv & 0 & \pmod{m_j} \quad \text{für } j \neq \ell
\end{array}$$

□

Beispiel 2.2.4.

Zu Lösen ist

$$\begin{array}{lll}
x \equiv 3 & \pmod{4} & \text{d.h.} \quad a_1 = 3 \quad m_1 = 4 \\
x \equiv 4 & \pmod{5} & \qquad\quad a_2 = 4 \quad m_2 = 5 \\
x \equiv 1 & \pmod{9} & \qquad\quad a_3 = 1 \quad m_3 = 9
\end{array}$$

Berechnen der Basis-Lösungen aus den **Moduln** m_i:

Modul	Prod. d. Restlichen	Bézout aus Euklid. Algo.	Basislösung
$m_1 = 4$	$M_1 = 5 \cdot 9 = \mathbf{45}$	$\underbrace{-11 \cdot 4}_{-44} + 1 \cdot 45 = 1 \Rightarrow$	$e_1 := \mathbf{45}$
$m_2 = 5$	$M_2 = 4 \cdot 9 = \mathbf{36}$	$\underbrace{-7 \cdot 5}_{-35} + 1 \cdot 36 = 1 \Rightarrow$	$e_2 := \mathbf{36}$
$m_3 = 9$	$M_3 = 4 \cdot 5 = \mathbf{20}$	$\underbrace{9 \cdot 9}_{81} + (-4) \cdot 20 = 1 \Rightarrow$	$e_3 := \mathbf{-80}$

Berechnen einer Lösung aus den Resten a_i und Basis-Lösungen e_i:
Eine der (unendlich vielen!) Lösungen ist:

$$\begin{aligned} \mathbf{x} &= 3 \cdot e_1 + 4 \cdot e_2 + 1 \cdot e_3 \\ &= 3 \cdot 45 + 4 \cdot 36 + 1 \cdot (-80) = \mathbf{199} \end{aligned}$$

Diese Lösung ist eindeutig modulo $m := 4 \cdot 5 \cdot 9 = 180$, da $180 < 199$ gilt, ist 199 nicht die kleinste positive Lösung, sondern

$$\mathsf{Rest}(199, 180) = 199 - 180 = 19 \quad \text{ist kleinste positive Lösung}$$

Die Menge aller Lösungen lautet: $\{199 + k \cdot 180 \ : \ k \in \mathbb{Z}\}$ bzw. $\{19 + k \cdot 180 \ : \ k \in \mathbb{Z}\}$.

2.2.1. Anwendung des Chinesischen Restsatzes: Probabilistischer Gleichheitstest

Wir möchten nun einen Anwendung des Chinesischen Restsatzes diskutieren. Dabei sind zwei große Zahlen zu vergleichen, indem nur wenige Bits tatsächlich verglichen werden.

► **Die Aufgabe:** Zwei Personen an den Enden eines Nachrichtenkanals wollen zwei 10 000-Bit lange, binäre Nachrichten auf Gleichheit hin überprüfen. Dabei sollen möglichst wenige Bits übertragen werden.

► **Der Algorithmus:** Das folgende Verfahren erlaubt einen Vergleich der beiden als Zahlen $a, b < 2^{10\,000}$ interpretierbaren Nachrichten, wobei anstatt der bis zu 10 000 Bits für die gesamte Nachricht nur $k \cdot 101$ Bits gesendet werden (bzw. $k \cdot 202$ Bits bei erforderlicher Übertragung der Moduln). Das Verfahren garantiert schon für $k = 1$ sehr hohe Sicherheit.

Algorithmus 2.2.5 (Probabilistischer Gleichheitstest)**.**

Input: $a, b \in \mathbb{N}$ mit $a, b < 2^{10\,000}$.
Output: „$a \neq b$“ oder „*mit großer Wahrscheinlichkeit* $a = b$“.

1. Wähle zufällig Primzahlen $p_1, \ldots, p_k \in [2^{100}, 2^{101}]$.
2. Übertrage a modulo p_i für alle $i = 1, \ldots, k$.

if $a \not\equiv b \pmod{p_i}$ für ein i **then**
  return „$a \neq b$“
else
  return „*mit großer Wahrscheinlichkeit* $a = b$“
end if

► **Die Analyse:** Ist die Ausgabe des Verfahrens „$a \neq b$“ so ist dies stets richtig, da aus $a \not\equiv b \pmod{p_i}$ (für eine der Primzahlen p_i) sofort $a \neq b$ folgt. Wir schätzen nun die Wahrscheinlichkeit ab, dass das Verfahren zu einer Fehlentscheidung der Form „$a = b$“ führt, obwohl $a \neq b$ gilt. Zentral ist dabei, dass es für $a \neq b$ nicht viele p_i geben kann mit $a \equiv b \pmod{p_i}$. Folgende Beobachtung hilft enorm bei der Analyse.

Lemma 2.2.6. Für zwei natürliche Zahlen $a, b \in \mathbb{N}$ mit $a \neq b$ gibt es höchstens 99 Primzahlen $p_i \in [2^{100}, 2^{101}]$ mit $a \equiv b \pmod{p_i}$.

Beweis. Wir beweisen die Aussage durch Widerspruch.

Annahme: Es gibt 100 verschiedene Primzahlen $q_1, \ldots q_{100} \in [2^{100}, 2^{101}]$ mit $a \equiv b \pmod{q_i}$ für alle $i = 1, \ldots 100$.

Nach dem Chinesischen Restsatz ist dann aber $a \equiv b \pmod{m}$ mit $m := q_1 \cdots q_{100}$. Es gilt nach der Wahl der Primzahlen aus dem Intervall $[2^{100}, 2^{101}]$, dass

$$m > (2^{100})^{100} = 2^{10\,000}.$$

Da jedoch $a, b < 2^{10\,000} < m$ sind sie identisch $a = b$. (Denn $x \equiv b \pmod{m}$ wird durch $x = b$ erfüllt und sonst keiner Zahle zwischen 0 und m.) □

Eine Fehlentscheidung ist also überhaupt nur dann möglich, falls das Verfahren zufälligerweise *ausschließlich* solche Primzahlen $q > 2^{100}$ auswählt für welche $a \equiv b \pmod{q}$ gilt. Von diesen gibt es höchstens 99. Wie groß die Wahrscheinlichkeit für dieses Ereignis ist, hängt von der Anzahl der Primzahlen in dem Intervall $[2^{100}, 2^{101}]$ ab. Mit Hilfe des berühmten Primzahlsatzes lässt sich dies Anzahl abschätzen.

Lemma 2.2.7. In dem Intervall $[2^{100}, 2^{101}]$ gibt es ungefähr 2^{93} Primzahlen.

Beweis. Nach dem berühmten Primzahlsatz gilt für die Anzahl $\pi(n)$ der Primzahlen in $[1, n]$ die asymptotische Formel $\pi(x) \simeq n/\ln(n)$. In $[2^{100}, 2^{101}]$ gibt es daher approximativ 2^{93} Primzahlen:

$$\begin{aligned}\pi\left(2^{101}\right) - \pi\left(2^{100}\right) &\simeq \frac{2^{101}}{\ln\left(2^{101}\right)} - \frac{2^{100}}{\ln\left(2^{100}\right)} \geq \frac{1}{\ln\left(2^{100}\right)}\left(2^{101} - 2^{100}\right) \\ &= \frac{2^{100}}{100 \cdot \ln(2)} \geq 2^{93}\end{aligned}$$

□

Lemma 2.2.8. Die Fehlerwahrscheinlichkeit, bei unterschiedlichen Inputzahlen $a, b < 2^{10\,000}$ ist nach oben beschränkt durch $\left(\frac{99}{2^{93}}\right)^k$.

Beweis. Nach Lemma 2.2.6 gibt es höchstens 99 „ungeeigneten“ Primzahlen in $[2^{100}, 2^{101}]$. Die Wahrscheinlichkeit zufällig eine von diesen zu erwischen ist nach Lemma 2.2.7 ungefähr $99/2^{93} \simeq 0,9996 \cdot 10^{-26}$. Bei k-facher unabhängiger Wahl einer Primzahl aus $[2^{100}, 2^{101}]$ ist die Fehlerwahrscheinlichkeit also höchstens $\left(\frac{99}{2^{93}}\right)^k$. □

Bemerkung 2.2.9.

Schon für $k = 1$ ist die Fehlerwahrscheinlichkeit des probabilistischen Gleichheitstest also verschwindend klein.

Man beachte: Diese Schranke für die Fehlerwahrscheinlichkeit ist unabhängig von den Nachrichten a und b. Wenn wir dagegen an zufälligen Bitpositionen prüfen, erkennen wir die Ungleichheit oft nicht, wenn a und b an fast allen Bitpositionen übereinstimmen.

2.3. Die Eulersche Phi-Funktion

Nachdem wir die Eulersche φ-Funktion erinnert haben, werden wir uns mit der Aufgabe beschäftigen, Urbilder der φ-Funktion zu finden. Die φ-Funktion spielt bei der späteren Behandlung von Anwendungen wie dem RSA-Schema eine zentrale Rolle.

Die eulersche φ-Funktion

Definition 2.3.1 (Eulersche φ-Funktion)**.** Für $n \in \mathbb{N}$ ist die Eulersche φ-Funktion definiert als

$$\varphi(n) := |\{m \in \{1, \ldots, n\} \ : \ \mathrm{ggT}(m, n) = 1\}|$$

Satz 2.3.2. Für eine Primzahl $p \in \mathbb{N}$ gilt:

- $\varphi(p) = (p - 1)$
- $\varphi(p^k) = p^{k-1}(p - 1)$ mit Exponenten $k \in \mathbb{N}$, $k \neq 0$.
- Ist $n = p_1^{k_1} \cdot p_2^{k_2} \cdots p_\ell^{k_\ell}$ die Primzahlzerlegung von n mit $p_1 < p_2 < \ldots < p_\ell$ und $k_1, k_2 \ldots, k_\ell \neq 0$, so gilt:

$$\varphi(n) = p_1^{k_1-1} \cdot (p_1 - 1) \ \cdot p_2^{k_2-1} \cdot (p_2 - 1) \ \cdots \ p_\ell^{k_\ell-1} \cdot (p_\ell - 1)$$

Lemma 2.3.3. Es seien $n, m \in \mathbb{N}$. Es gilt genau dann $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$, wenn $\mathrm{ggT}(m, n) = 1$.

Satz 2.3.2 liefert nun die Grundlage, um $\varphi(n)$ zu berechnen: Zunächst muss n in seine Primfaktorzerlegung zerlegt werden, und dann kann mittels Regel iii) die Berechnung von $\varphi(n)$ erfolgen.

Wir möchten nun die umgekehrte Frage stellen. Lässt sich zu einem gegebenen m entscheiden, welche natürlichen Zahlen durch die Eulersche φ-Funktion auf m abgebildet wird.

2.3.1. Rückwärtsberechnung von φ

In diesem Abschnitt widmen wir uns der folgenden Aufgabe:

Gegeben ist eine Zahl $m \in \mathbb{N}$ und gesucht sind alle Zahlen n mit $\varphi(n) = m$.

Dabei gibt es (vermutlich) entweder *keine Lösung* oder aber *mehr als eine Lösung*.

Exkurs 2.3.4 (Anzahl Lösungen von $\varphi(n) = m$)**.**

Im Jahr 1922 stellte R.D. Carmichael bereits die Vermutung auf, das es für jedes $n \in \mathbb{N}$ mindestens eine weitere Zahl $k \in \mathbb{N}$, $k \neq n$ gibt mit $\varphi(n) = \varphi(k)$. Tatsächlich hatte Carmichael im Jahr 1907 zunächst einen – falschen – Beweis für diese Aussage veröffentlicht.

Unter massivem Computereinsatz zeigten A. Schlafly und S. Wagon (1994), dass die Aussage zumindestens bis zu der (extrem großen) Zahl von $10^{(10^7)}$ richtig ist. Durch Erweiterung dieses Resultats verbesserte K. Ford diese untere Schranke auf $10^{\left(10^{10}\right)}$ (Ann. Math. 150:283–311, 1999).

Wie prüft man systematisch für eine Zahl $m \in \mathbb{N}$, ob sie im Wertebereich der eulerschen φ-Funktion liegt? Zunächst beobachten wir, dass nur gerade Zahlen und die 1 in Frage kommen.

Lemma 2.3.5. Sei $m \in \mathbb{N}$ und $m \geq 2$. Dann ist m gerade, wenn es ein $n \in \mathbb{N}$ gibt mit $\varphi(n) = m$. Außerdem ist $\varphi(n) = 1$ nur für $n = 1, 2$ erfüllt.

Beweis. Sei $m \in \mathbb{N}$. Wir nehmen an, es gibt ein $n \in \mathbb{N}$ so dass $\varphi(n) = m$ ist.

Wir unterscheiden zwei Fälle.

- **Fall 1:** $n \geq 3$**.**
  - **Fall 1.1:** $n = 2^k$**.** Nach Satz 2.3.2 gilt

$$m = \varphi(n) = \varphi(2^k) = 2^{k-1}(2-1). \tag{2.1}$$

    Nach Voraussetzung ist $n \geq 3$ und somit $k \geq 2$. Also ist mit (2.1) $m = \varphi(n)$ gerade und $m \neq 1$.
  - **Fall 1.2:** $n = 2^k \cdot n'$ **mit** $2 \nmid n'$ **und** $n' > 1$**.** In der Primfaktorzerlegung von n' taucht die Potenz von mindestens einer ungerade Primzahl p auf so dass $n = p^k \cdot n''$ mit $p \nmid n''$. Nach Satz 2.3.2 gilt

$$m = \varphi(n) = \varphi(p^k) \cdot \varphi(n'') = p^{k-1}(p-1) \cdot \varphi(n'') \tag{2.2}$$

    Da p ungerade ist, ist $p - 1$ gerade und mit 2.2 auch $\varphi(n)$ und somit m.
    Außerdem ist $p \geq 3$, also $(p-1) \geq 2$ und mit 2.2 $m \geq 2$ also insbesondere $m \neq 1$.
- **Fall 2:** $n < 3$**.** Es bleiben die beiden Möglichkeiten $n = 1$ oder $n = 2$. In beiden Fällen prüft man ganz leicht $\varphi(n) = 1$.

□

Bemerkung 2.3.6.

Das Lemma 2.3.5 impliziert nicht, dass es für ***jede*** gerade Zahl m tatsächlich auch eine natürliche Zahl $n \in \mathbb{N}$ gibt, so dass $\varphi(n) = m$. Beispielsweie gibt es keine natürliche Zahl die durch die eulersche φ-Funktion auf die 14 abgebildet wird, wie wir unten sehen werden.

Der Fall $m = 1$ ist durch Lemma 2.3.5 gelöst.

Für gerade $m \geq 2$ suchen wir alle Zahlen n mit $\varphi(n) = m$. Wir erreichen das Ziel nach der folgenden Definition in vier Schritten.

Definition 2.3.7 (Reine Lösung)**.** Sei $m \in \mathbb{N}$. Wir sagen $n \in \mathbb{N}$ ist eine ***reine Lösung*** für m wenn $\varphi(n) = m$ und $n = p^k$ für eine Primzahl p und $k \in \mathbb{N}$ ist.

Beispiel 2.3.8.

Für $m = 100$ sind die reinen Lösungen 101^1 und $5^3 = 125$ und es gilt $\varphi(101) = 100$ und $\varphi(5^3) = 5^2 \cdot (5-1) = 100$.

Schema F zur Rückwärtsberechnung von φ

Um die Urbilder bezüglich der eulerschen φ-Funktion einer natürlichen Zahl $m \in \mathbb{N}$ zu ermitteln, bedient man sich ihrer Multiplikativität. Das heißt um nicht alle Zahlen von 1 bis n untersuchen zu müssen hilft die Tatsache, dass wenn eine Zahl n auf m abgebildet wird, sich der Funktionswert (also $\varphi(n) = m$) in ein Produkt der Bilder der Primfaktoren aus der Primfaktorzerlegung von n zerlegen lässt. Ist nämlich

$$n = \prod_{i=1}^{t} p_i^{k_i} \qquad \Longrightarrow \qquad \varphi(n) = \prod_{i=1}^{t} \varphi(p_i^{k_i}) = \prod_{i=1}^{t} a_i$$

wobei die a_i nach Lemma 2.3.5 gerade oder 1 sind. Der letzte Fall tritt nur ein, wenn n durch 2 aber nicht durch 4 teilbar ist. Außerdem sind die Primfaktoren $p_i^{k_i}$ reine Lösungen der a_i. Man geht also zunächst auf die Sache nach Faktorisierungen von m in gerade Faktoren und sucht dann teilerfremde reine Lösungen, welche auf die Faktoren abbilden.

Wir geben jetzt drei Schritte an, welche dieses Programm zum Auffinden aller Lösungen für die Rückwärtsberechnung der φ-Funktion umsetzen. Im Anschluss werden wir beweisen, dass das Anwenden dieser Schritte zu korrekten Lösungen führt und auch jede korrekte Lösung gefunden wird.

► **Schritt 1.** Finde alle Faktorisierungen (Zerlegungen) von m mit ausschließlich geraden Faktoren.

► **Schritt 2.** Es werden für jede Faktorisierung $m = a_1 \cdot a_2 \cdots a_\ell$ aus Schritt 1 die folgenden Schritte durchgeführt.

- ► **Schritt 2.1.** finde für alle Faktoren a_i die Menge aller ***reinen Lösungen***. Es kommen jeweils nur zwei Zahlen in Frage:
  - Die Zahl $a_i + 1$ ist eine reine Lösung von a_i, wenn sie eine Primzahl ist. Dann sei $b_i^{[1]} = a_i + 1$
  - Für die größte Primzahl p_1 aus der Primfaktorzerlegung von $a_i = \prod_{j=1}^{\ell} p_j^{k_j}$ prüft man ob

$$\prod_{j=2}^{s} p_j^{k_j} = (p_1 - 1). \tag{2.3}$$

Ist dies der Fall, ist $b_i^{[2]} = p_1^{k_1+1}$ eine reine Lösung von a_i.

Gibt es für einen Faktor keine reine Lösung, dann wird die Faktorisierung $m = a_1 \cdot a_2 \cdots a_\ell$ verworfen und Schritt 2.2 nicht ausgeführt.

► **Schritt 2.2.** Nun werden je paarweise teilerfremde reine Lösungen der Faktoren miteinander multipliziert. Sei

$$b^{[s_1,s_2,\ldots,s_\ell]} = b_1^{[s_1]} \cdot b_2^{[s_2]} \cdots b_\ell^{[s_\ell]}$$

wobei die $b_i^{[s_i]}$ je eine reine Lösung des Faktors a_i sind.

► **Schritt 3.** Für jede gefundene ungerade Lösung b ist auch $2b$ eine Lösung.

Die drei Schritte in algorithmischer Schreibweise.

Algorithmus 2.3.9 (Auffinden aller Lösungen von $\varphi(n) = m$)**.**

Input: $m \in \mathbb{N}$.
Output: Alle Zahlen n mit $\varphi(n) = m$.
1. Setze $\mathbb{L} := \{\}$.
2. Bestimme alle möglichen Zerlegungen $m_u = a_{u1} \cdots a_{u\ell(u)}$ von m in *gerade* Faktoren a_{ui}.
3. Sei d die Anzahl aller Zerlegungen aus 2.
for $u = 1$ **to** d **do**
  for $i = 1$ **to** $\ell(u)$ **do**
    4. Berechne für a_{ui} die Primfaktorzerlegung von $a_{ui} = \prod_{j=1}^{v} p_j^{k_j}$
    if $a_{ui} + 1$ ist eine Primzahl **then**
      Dann setze $b_i^{[1]} = a_{ui} + 1$
    else if $\prod_{j=1}^{v} p_j^{k_j} = p_1 - 1$ **then**
      Dann setze $b_i^{[2]} = p_1^{k_1+1}$
    else
      Setze u=u+1 und Exit FOR
    end if
  end for
  5. Für alle Tupel $(b_1^{[s_1]}, \ldots, b_\ell^{[s_\ell]})$ mit *paarweise verschiedenen* $b_i^{[s_i]}$ füge Lösung in $\mathbb{L}$ ein:
    $\mathbb{L} := \mathbb{L} \bigcup \{b_1^{[s_1]} \cdots b_\ell^{[s_\ell]}\}$
end for
6. Für jede *ungerade* Lösung $n \in \mathbb{L}$ füge Lösung $2n$ in $\mathbb{L}$ ein:
  $\mathbb{L} := \mathbb{L} \bigcup \{2 \cdot n \ : \ n \in \mathbb{L}\,,\ n \text{ ungerade}\}$
return $\mathbb{L}$.

Wir beweisen nun, dass der Algorithmus 2.3.9 alle und nur korrekte Lösungen findet. Dazu betrachten wir die durchgeführten Schritte zunächst einzeln.

Lemma 2.3.10. Schritt 1 findet alle potenziellen Faktorisierungen von m für welche es teilerfremde Zahlen $b_1, \ldots, b_s$ geben kann, deren φ-Funkionswert auf die Faktoren der Faktorisierung abbildet.

Beweis. Dieser erste Schritt motiviert sich aus der Beobachtung von Lemma 2.3.3. Für Zahlen $b_1, b_2, \ldots, b_s$ mit $n = b_1 \cdot b_2 \cdots b_s$ gilt

$$\varphi(n) = \varphi(b_1) \cdot \varphi(b_2) \cdots \varphi(b_s) \qquad \Leftrightarrow \qquad \mathrm{ggT}(b_i, b_j) = 1 \quad \forall\, i, j \text{ mit } 1 \le i < j \le s.$$

Für jede Faktorisierung von $m = a_1 \cdot a_2 \cdots a_\ell$ reduziert sich das Problem ein $n \in \mathbb{N}$ zu finden mit $\varphi(n) = m$ auf das Problem paarweise teilerfremde $b_1, b_2, \ldots, b_s \in \mathbb{N}$ zu finden mit $\varphi(b_i) = a_i$ für alle $1 \leq i \leq s$.

Faktorisierungen mit mindestens einem ungeraden a_i können wir dabei ausschließen, es gibt nach Lemma 2.3.5 sicherlich kein $b_i \in \mathbb{N}$ mit $\varphi(b_i) = a_i$. □

Lemma 2.3.11. Schritt 2.1 findet für jeden Faktor aus einer fixierten Faktorisierung $a_1 \cdot a_2 \cdots a_\ell$ aus Schritt 1 die Menge aller reinen Lösungen.

Beweis. Erfüllt die größte Primzahl p_1 aus der Primfaktorzerlegung von a_i die Gleichung (2.3), dann gilt

$$\varphi(p_1^{k_1+1}) = p_1^{k_1} \cdot (p_1 - 1) = p_1^{k_1} \cdot \prod_{j=2}^{\ell} p_j^{k_j} = \prod_{j=1}^{\ell} p_j^{k_j} = a_i.$$

Man prüft das alleine für die größte Primzahl aus der Faktorisierung, denn schon für die zweitgrößte Primzahl p_2 in der Primfaktorzerlegung von a_i gilt, dass

$$\prod_{\substack{j=1 \\ j \neq 2}}^{\ell} p_j^{k_j} = p_1 \cdot \prod_{j=3}^{\ell} p_j^{k_j} > p_2 - 1$$

und somit

$$\varphi(p_2^{k_2+1}) = p_2^{k_2} \cdot (p_2 - 1) \neq p_2^{k_2} \cdot \prod_{\substack{j=1 \\ j \neq 2}}^{\ell} p_j^{k_j} = \prod_{j=1}^{\ell} p_j^{k_j} = a_i.$$

Für eine Primzahl $m + 1$ ist $\varphi(m + 1) = (m + 1 - 1) = m$. □

Lemma 2.3.12. Sei $a_1 \cdot a_2 \cdots a_\ell$ eine fixierte Faktorisierung aus Schritt 1, welche nicht in Schritt 2.1 verworfen wurde. Die in Schritt 2.2 Produkte von reinen Lösungen werden auf m abgebildet. Eine in Schritt 3 gefundene Zahl wird auch ebenfalls auf m abgebildet.

Beweis. Sei $b^{[s_1,s_2,\ldots,s_\ell]} = b_1^{[s_1]} \cdot b_2^{[s_2]} \cdots b_\ell^{[s_\ell]}$ eine in Schritt 2.2 gebildete Lösung. Da die $b_i^{[s_i]}$ paarweise teilerfremd sind, gilt für eine solche Lösung

$$\varphi(b^{[s_1,s_2,\ldots,s_\ell]}) = \varphi(b_1^{[s_1]} \cdot b_2^{[s_2]} \cdots b_\ell^{[s_\ell]}) = \varphi(b_1^{[s_1]}) \cdot \varphi(b_2^{[s_2]}) \cdots \varphi(b_\ell^{[s_\ell]}) = a_1 \cdot a_2 \cdots a_\ell = m.$$

Sei b eine ungerade gefundene Lösung. Es gelte also $\varphi(b) = m$ und 2 und b sind teilerfremd. Dann gilt aber $\varphi(2 \cdot b) = \varphi(2) \cdot \varphi(b) = 1 \cdot \varphi(b) = m$. □

Lemma 2.3.13. Es werden alle Lösungen gefunden.

Beweis. Sei $n \in \mathbb{N}$ eine Zahl mit $\varphi(n) = m$. Wir prüfen, ob diese Zahl mit Hilfe der vier Schritte gefunden wird. Betrachtet man die Primfaktorzerlegung von $n = 2^k \cdot \prod_{i=1}^{t} p_i^{k_i}$ in das Produkt paarweise verschiedener ungerader Primzahlen p_i für $i = 1, \ldots, t$ und einer Potenz von 2. Dann gibt es drei Fälle.

- ***Es ist*** $k = 0$. Dann ist n ungerade. Es ist $\varphi(n) = \prod_{i=1}^{t} \varphi(p_i^{k_i}) = \prod_{i=1}^{t} a_i$ wobei $p_i^{k_i}$ jeweils eine reine

Lösung zu a_i ist. Die Faktoren a_i sind nach Lemma 2.3.5 alle gerade. Die Zahl n wird also bezüglich der Faktorisierung $\prod_{i=1}^{t} a_i = m$ gefunden.

- ***Es ist*** $k = 1$**.** Dann ist $\frac{n}{2}$ ungerade. Es ist außerdem $\varphi(n) = \varphi(2) \cdot \prod_{i=1}^{t} \varphi(p_i^{k_i}) = \prod_{i=1}^{t} \varphi(p_i^{k_i}) = \varphi(\frac{n}{2})$ und $\prod_{i=1}^{t} \varphi(p_i^{k_i}) = \prod_{i=1}^{t} a_i$ wobei $p_i^{k_i}$ jeweils eine reine Lösung zu a_i ist. Die Faktoren a_i sind nach Lemma 2.3.5 alle gerade. Die Zahl $\frac{n}{2}$ wird also bezüglich der Faktorisierung $\prod_{i=1}^{t} a_i = m$ gefunden. Da $\frac{n}{2}$ ungerade ist, findet man n im Schritt 3, in welchem alle ungeraden Lösungen mit 2 multipliziert werden.
- ***Es ist*** $k \geq 2$**.** Es ist $\varphi(n) = 2^k \cdot \prod_{i=1}^{t} \varphi(p_i^{k_i}) = 2^{k-1} \cdot \prod_{i=0}^{t} a_i$ wobei $p_i^{k_i}$ jeweils eine reine Lösung zu a_i und 2^k eine reine Lösung zu 2^{k-1} ist. Die Faktoren a_i und 2^{k-1} sind nach Lemma 2.3.5 und weil $k \geq 2$ ist alle gerade. Die Zahl n wird also bezüglich der Faktorisierung $2^{k-1} \cdot \prod_{i=1}^{t} a_i = m$ gefunden.

□

Beispiel 2.3.14.

Wir betrachten zunächst nur die Schritte 2.1 und 2.2 für $m = 400$ und die Zerlegung $m = 4 \cdot 100$.

Finde reine Lösungen für $a_1 = 4$.

$4 \xrightarrow{+1} 5$ ✓ - Primzahl. $\rightarrow 5$ ist reine Lösung.

$\downarrow$ faktorisieren

$2^2 \longrightarrow 2 \rightarrow 1 = (2-1)$ ✓ $\rightarrow 2^3$ ist reine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[1]} = 5}$ und $\mathbf{b_1^{[2]} = 8}$

Finde reine Lösungen für $a_2 = 100$.

$100 \xrightarrow{+1} 101$ ✓ - Primzahl. $\rightarrow 101$ ist reine Lösung.

$\downarrow$ faktorisieren

$5^2 \cdot 2^2 \longrightarrow 5 \rightarrow 2^2 = (5-1)$ ✓ $\rightarrow 5^3$ ist reine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[1]} = 101}$ und $\mathbf{b_1^{[2]} = 125}$

Wir kombinieren die reinen Lösungen

$$b^{[1,1]} = b_1^{[1]} \cdot b_2^{[1]} = 5 \cdot 101 = 505$$

$$b^{[2,1]} = b_1^{[2]} \cdot b_2^{[1]} = 8 \cdot 101 = 808$$

$$b^{[1,2]} = b_1^{[1]} \cdot b_2^{[2]} = 5 \cdot 125 \quad \text{↯ - nicht teilerfremd}$$

$$b^{[2,2]} = b_1^{[2]} \cdot b_2^{[2]} = 8 \cdot 125 = 1000$$

und erhalten drei Lösungen für die Zerlegung $400 = 4 \cdot 100$.

Beispiel 2.3.15.

Gegeben ist $m = 20$, gesucht sind alle Lösungen n von $\varphi(n) = 20$.

► **Schritt 1.** Es gilt: $m = 20 = 2 \cdot 10$, d.h. m hat zwei Zerlegungen in gerade Faktoren "20" und "$2 \cdot 10$".

Für jede Zerlegung führe Schritte 2.1 und 2.2 durch.

Zerlegung $m = 20$.

► **Schritt 2.1**

Finde reine Lösungen für $a_1 = 20$.

$20 \xrightarrow{+1} 21 = 3 \cdot 7$ ↯ - keine Primzahl. $\rightarrow$ 21 ist keine Lösung.

$\downarrow$ faktorisieren

$5 \cdot 2^2 \longrightarrow 5 \rightarrow 2^2 = (5-1)$ ✓ $\rightarrow 5^2$ ist reine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[1]} = 25}$

► **Schritt 2.2** Entfällt an dieser Stelle, die Faktorisierung besteht nur aus einem Faktor.

Zerlegung $m = 2 \cdot 10$.

► **Schritt 2.1**

Finde reine Lösungen für $a_1 = 2$.

$2 \xrightarrow{+1} 3$ ✓ - Primzahl. $\rightarrow 3$ ist reine Lösung.

$\downarrow$ faktorisieren

$2 \longrightarrow 2 \rightarrow 1 = (2-1)$ ✓ $\rightarrow 2^2$ ist reine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[1]} = 3}$ und $\mathbf{b_1^{[2]} = 2^2}$

Finde reine Lösungen für $a_2 = 10$.

$10 \xrightarrow{+1} 11$ ✓ - Primzahl. $\rightarrow 11$ ist reine Lösung.

$\downarrow$ faktorisieren

$5 \cdot 2 \longrightarrow 5 \rightarrow 2 \neq (5-1)$ ↯ $\rightarrow 5^2$ ist keine Lösung.

Reine Lösungen sind: $\mathbf{b_2^{[1]} = 11}$

► **Schritt 2.2** Lösungen für Zerlegung $m = 2 \cdot 10$ sind ...

$\mathbf{b^{[1,1]} = b_1^{[1]} \cdot b_2^{[1]} = 3 \cdot 11 = 33}$ ✓ - weil 3 und 11 teilerfremd sind

$\mathbf{b^{[2,1]} = b_1^{[2]} \cdot b_2^{[1]} = 4 \cdot 11 = 44}$ ✓ - weil 4 und 11 teilerfremd sind

Zusammentragen der bisherigen Lösungen:

$\mathbb{L} = \{25, 33, 44\}$

► **Schritt 3.** Für jede gefundene ungerade Lösung n füge $2n$ hinzu:

$\mathbb{L} = \{25, 2 \cdot 25,\ 33, 2 \cdot 33,\ 44\} = \{25, 50, 33, 66, 44\}$

Beispiel 2.3.16.

Gegeben ist $m = 14$, gesucht sind alle Lösungen n von $\varphi(n) = 14$.

► **Schritt 1.** Es gilt: $m = 14$, d.h. m hat genau eine Zerlegungen in gerade Faktoren "14" selbst.

Für diese Zerlegung führen wir nun Schritte 2.1 und 2.2 durch.

Zerlegung $m = 14$.

► **Schritt 2.1.**

Finde reine Lösungen für $a_1 = 14$.

$14 \xrightarrow{+1} 15 = 3 \cdot 5$ ↯ - keine Primzahl. $\rightarrow$ 15 ist keine Lösung.

$\downarrow$ faktorisieren

$7 \cdot 2 \longrightarrow 7 \rightarrow 2 \neq (7 - 1)$ ↯ $\rightarrow 7^2$ ist keine Lösung.

Reine Lösungen sind: $\emptyset$

► **Schritt 2.2.** Entfällt an dieser Stelle, die Faktorisierung besteht nur aus einem Faktor, außerdem haben wir keine reinen Lösungen gefunden.

► **Schritt 3.** Dieser Schritt entfällt, es sind keine Lösungen gefunden worden.

Beispiel 2.3.17.

Gegeben ist $m = 40$, gesucht sind alle Lösungen n von $\varphi(n) = 40$.

► **Schritt 1.** Es gilt: $m = 40 = 4 \cdot 10 = 2 \cdot 20 = 2 \cdot 2 \cdot 10$, d.h. m hat 4 Zerlegungen in gerade Faktoren.

Für jede Zerlegung und jeden Faktor suche die reinen Lösungen:

Zerlegung $m = 40$

► **Schritt 2.1.**

Finde reine Lösungen für $a_1 = 40$.

$40 \xrightarrow{+1} 41$ ✓ - Primzahl. $\rightarrow 41$ ist reine Lösung.

$\downarrow$ faktorisieren

$5 \cdot 2^3 \longrightarrow 5 \rightarrow 1 = (2-1)$ ↯ $\rightarrow 5^2$ ist keine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[1]} = 41}$

► **Schritt 2.2.** Entfällt an dieser Stelle, die Faktorisierung besteht nur aus einem Faktor.

Zerlegung $m = 4 \cdot 10$

► **Schritt 2.1.**

Finde reine Lösungen für $a_1 = 4$.

$4 \xrightarrow{+1} 5$ ✓ - Primzahl. $\rightarrow 5$ ist reine Lösung.

$\downarrow$ faktorisieren

$2^2 \longrightarrow 2 \rightarrow 1 = (2-1)$ ✓ $\rightarrow 2^3$ ist reine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[1]} = 5}$ und $\mathbf{b_1^{[2]} = 8}$

Finde reine Lösungen für $a_2 = 10$.

$10 \xrightarrow{+1} 11$ ✓ - Primzahl. $\rightarrow 11$ ist reine Lösung.

$\downarrow$ faktorisieren

$5 \cdot 2 \longrightarrow 5 \rightarrow 2 \neq (5-1)$ ↯ $\rightarrow 5^2$ ist keine Lösung.

Reine Lösungen sind: $\mathbf{b_2^{[1]} = 11}$

► **Schritt 2.2.** Lösungen für Zerlegung $m = 4 \cdot 10$ sind

$\mathbf{b^{[1,1]} = b_1^{[1]} \cdot b_2^{[1]} = 5 \cdot 11 = 55}$ ✓ - weil 5 und 11 teilerfremd sind

$\mathbf{b^{[2,1]} = b_1^{[2]} \cdot b_2^{[1]} = 8 \cdot 11 = 88}$ ✓ - weil 8 und 11 teilerfremd sind

Zerlegung $m = 2 \cdot 20$

► **Schritt 2.1.**

Finde reine Lösungen für $a_1 = 2$.

$2 \xrightarrow{+1} 3$ ✓ - Primzahl. $\rightarrow 3$ ist reine Lösung.

$\downarrow$ faktorisieren

$2 \longrightarrow \quad 2 \rightarrow 1 = (2-1)$ ✓ $\rightarrow 2^2$ ist reine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[1]} = 3}$ und $\mathbf{b_1^{[2]} = 2^2}$

Finde reine Lösungen für $a_1 = 20$.

$20 \xrightarrow{+1} 21 = 3 \cdot 7$ ↯ - keine Primzahl. $\rightarrow 21$ ist keine Lösung.

$\downarrow$ faktorisieren

$5 \cdot 2^2 \longrightarrow \quad 5 \rightarrow 2^2 = (5-1)$ ✓ $\rightarrow 5^2$ ist reine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[2]} = 25}$

► **Schritt 2.2.** Lösungen für Zerlegung $m = 2 \cdot 20$ sind

$\mathbf{b^{[1,2]} = b_1^{[1]} \cdot b_2^{[2]} = 3 \cdot 25 = 75}$ ✓ - weil 3 und 25 teilerfremd sind

$\mathbf{b^{[2,2]} = b_1^{[2]} \cdot b_2^{[2]} = 4 \cdot 25 = 100}$ ✓ - weil 4 und 25 teilerfremd sind

Zerlegung $m = 2 \cdot 2 \cdot 10$

► **Schritt 2.1.**

Finde reine Lösungen für $a_1 = a_2 = 2$.

$2 \xrightarrow{+1} 3$ ✓ - Primzahl. $\rightarrow 3$ ist reine Lösung.

$\downarrow$ faktorisieren

$2 \longrightarrow 2 \rightarrow 1 = (2-1)$ ✓ $\rightarrow 2^2$ ist reine Lösung.

Reine Lösungen sind: $\mathbf{b_1^{[1]} = b_2^{[1]} = 3}$ und $\mathbf{b_1^{[2]} = b_2^{[2]} = 2^2}$

Finde reine Lösungen für $a_2 = 10$.

$10 \xrightarrow{+1} 11$ ✓ - Primzahl. $\rightarrow 11$ ist reine Lösung.

$\downarrow$ faktorisieren

$5 \cdot 2 \longrightarrow 5 \rightarrow 2 \neq (5-1)$ ↯ $\rightarrow 5^2$ ist keine Lösung.

Reine Lösungen sind: $\mathbf{b_2^{[2]} = 11}$

► **Schritt 2.2.** Lösungen für Zerlegung $m = 2 \cdot 2 \cdot 10$ sind

~~$\mathbf{b^{[1,1,2]}}$~~ $= \mathbf{b_1^{[1]} \cdot b_2^{[1]} \cdot b_3^{[2]} = 3 \cdot 3 \cdot 11 = \cancel{99}}$ ↯ - weil 3 und 3 nicht teilerfremd sind

$\mathbf{b^{[2,1,2]} = b_1^{[2]} \cdot b_2^{[1]} \cdot b_3^{[2]} = 4 \cdot 3 \cdot 11 = 132}$ ✓ - weil $3, 4$ und 11 teilerfremd sind

~~$\mathbf{b^{[1,2,2]}}$~~ $= \mathbf{b_1^{[1]} \cdot b_2^{[1]} \cdot b_3^{[2]} = 3 \cdot 4 \cdot 11 = \cancel{132}}$ ✓ - weil $3, 4$ und 11 teilerfremd sind - aber gleich $\mathbf{b^{[2,1,1]}}$

~~$\mathbf{b^{[2,2,2]}}$~~ $= \mathbf{b_1^{[1]} \cdot b_2^{[1]} \cdot b_3^{[2]} = 4 \cdot 4 \cdot 11 = \cancel{176}}$ ↯ - weil 4 und 4 nicht teilerfremd sind

Zusammentragen der bisherigen Lösungen:

$\{41,\ 55, 88,\ 75, 100,\ 132\}$

► **Schritt 3.** Für jede gefundene ungerade Lösung n füge $2n$ hinzu:

$\mathbb{L} = \{41,\ 42,\ 55,\ 110,\ 88,\ 75,\ 150,\ 100, 132\}$

3 Kryptographie

3.1. Der Satz von Euler

In diesem Abschnitt erinnern wir zunächst die multiplikativen Gruppe $(\mathbb{Z}_n^*, \odot_n)$ (kurz $\mathbb{Z}_n^*$ genannt). Diese Gruppe spielt eine wichtige Rolle im Herleiten der Sätze, die wir für das "RSA-Schema" benötigen (ein Verschlüsselungsverfahren aus der Kryptographie).

Mit Hilfe des Satzes von Bézout lassen sich über den erweiterten euklidischen Algorithmus Inverse in $\mathbb{Z}_n^*$ berechnen. Wir möchten uns nun noch damit beschäftigen, wie Potenzen der Form

$$a^k := a \odot_n a \odot_n \cdots \odot_n a \qquad \text{für } a \in \mathbb{Z}_n^*$$

berechnet werden können. Die Bebachtungen werden wir in dem Satz von Fermat und dem Satz von Euler festhalten.

Erinnerung

Wir erinnern uns für $n \in \mathbb{N}$ an die Definition der Menge

$$\mathbb{Z}_n^* := \{k \in \mathbb{N} \ : \ k \leq n \ \text{ mit } \ \mathrm{ggT}(k, n) = 1\}$$

der teilerfremden Zahlen zu n. Schon bekannt ist, dass $(\mathbb{Z}_n^*, \odot_n)$ eine abelsche Gruppe bildet.

Satz 3.1.1. Es sei $n \in \mathbb{N}$ und $n \geq 2$, dann ist $(\mathbb{Z}_n^*, \odot_n)$ eine abelsche Gruppe.

Will man für ein $a \in \mathbb{Z}_n^*$ das passende Inverse $\overline{a}$ berechnen, so führt man folgende Berechnungsvorschrift aus:

1. Berechne via erweitertem Euklidischen Algorithmus Bézout-Multiplikatoren $s, t \in \mathbb{Z}$ mit $s \cdot a + t \cdot n = \mathrm{ggT}(a, n) = 1$.
2. Berechne $\overline{a} = \mathsf{Rest}(s, n)$.

Satz 3.1.2. Es sei $(G, \circ)$ eine endliche abelsche Gruppe mit neutralem Element e.

Für alle $a \in G$ gilt dann: $a^{|G|} = e$ ($|G|$ = Anzahl der Elemente von G).

Im Folgenden werden wir den Satz von Euler beweisen. Hierzu benötigen wir die Eigenschaften der Gruppe $\mathbb{Z}_n^*$, genauer gesagt rechnen wir mit $|\mathbb{Z}_n^*|$, der Anzahl der Elemente von $\mathbb{Z}_n^*$. Die Anzahl kennen wir schon, sie entspricht dem Wert der Eulerschen φ-Funktion von n.

3.1.1. Der Satz von Euler und der kleine Fermat

Wir wenden nun den Satz 3.1.2 auf Gruppen der Form $(\mathbb{Z}_n^*, \odot_n)$ an, und erhalten den Satz von Euler und den kleinen Satz von Fermat. Das folgende Korollar folgt direkt aus Satz 3.1.2 und der Tatsache, dass $(\mathbb{Z}_n^*, \odot_n)$ eine endliche abelsche Gruppe ist:

Korollar 3.1.3. Es sei $n \in \mathbb{N}$ und $n \geq 2$. In der abelschen Gruppe $(\mathbb{Z}_n^*, \odot_n)$ gilt $a^{\varphi(n)} = 1$ für jedes $a \in \mathbb{Z}_n^*$.

Aus diesem Korollar folgern wir den Satz von Euler:

Satz 3.1.4 (Satz von Euler)**.** Sind $k, n \in \mathbb{N}$ teilerfremd (d.h. $\text{ggT}(k,n) = 1$) mit $n \geq 2$ so folgt: $k^{\varphi(n)} \equiv 1 \pmod{n}$.

Satz 3.1.5 (kleiner Satz von Fermat)**.** Sind $k, p \in \mathbb{N}$ und p ist Primzahl so folgt: $k^p \equiv k \pmod{p}$.

Beweis. Es seien $k, p \in \mathbb{N}$ und es sei p eine Primzahl. Es gibt zwei Fälle zu untersuchen:

1. Gilt $\text{ggT}(k,p) = 1$ so folgt wegen $\varphi(p) = p - 1$ aus dem Satz von Euler
$$k^p = k^{p-1} \cdot k \equiv 1 \cdot k \pmod{p}.$$
2. Gilt $\text{ggT}(k,p) \neq 1$ so folgt sofort $\text{ggT}(k,p) = p$ (weil p nur die Teiler 1 und p hat, kommt als gemeinsamer Teiler von k und p nur noch p in Frage).
Es gilt also $k = \ell \cdot p$ mit $\ell \in \mathbb{N}$ und damit ist auch $k^p = \ell^p \cdot p^p$ ein Vielfaches von p, dh. es gilt $\text{Rest}(k,p) = 0 = \text{Rest}(k^p, p)$ bzw. $k^p \equiv k \pmod{p}$.

□

Bemerkung 3.1.6.

Der kleine Satz von Fermat stellt keine Bedingung an $k \in \mathbb{N}$, im Gegensatz zum Satz von Euler bei dem $\text{ggT}(k,n) = 1$ gelten muss ($k^{\varphi(n)} \equiv 1 \pmod{n}$ falls $\text{ggT}(k,n) = 1$). Dafür muss aber der Modul (die Zahl p) eine Primzahl sein.

Im Beweis wurde ersichtlich, dass k und p entweder teilerfremd sind - dann folgt der kleine Satz von Fermat direkt aus dem Satz von Euler - oder k ein Vielfaches von p ist - dann ist $k \equiv 0 \pmod{p}$ und die Aussage trivial.

Beispiel 3.1.7 (Satz von Euler)**.**

1. Für die Zahlen 3 und 5 gilt $\text{ggT}(3,5) = 1$.
Es gilt mit $\varphi(5) = 4$ also gilt nach Satz von Euler (Satz 3.1.4):
$3^4 = 81 \equiv 1 \pmod{5}$. Dies hätte man jedoch auch leicht selbst schließen können.

2. Mit etwas größeren Zahlen wird die Sache etwas schwieriger:
   Für 7 und $22 = 2 \cdot 11$ gilt $\mathrm{ggT}(7, 22) = 1$. Es gilt mit $\varphi(22) = 1 \cdot 10 = 10$ also
   $7^{10} = 282475249 \equiv 1 \pmod{22}$ bzw. $\mathsf{Rest}(7^{10}, 22) = 1$

Beispiel 3.1.8.

In Satz 3.1.2 ist die Bedingung $\mathrm{ggT}(k, n) = 1$ immens wichtig. Wählt man beispielsweise nicht-teilerfremde Zahlen $k = 2$ und $n = 8$, so gilt:

$$\varphi(8) = \varphi(2^3) = 2^2 \cdot 1 \qquad \text{und} \qquad 2^4 = 16 \equiv 0 \pmod{8}.$$

3.2. Schnelles Potenzieren

Wir werden nun mit Hilfe des Satzes von Euler eine Methode kennen lernen, wie man in Modulgleichungen schnell potenzieren kann und dann eine Methode, wie man Potenzieren mit großen Exponenten effizienter gestalten kann - die Anzahl der benötigten Multiplikationen reduzieren kann.

3.2.1. Schnelles Potenzieren in Modulgleichungen

Bevor wir die allgemeine Aussage in Lemma 3.2.2 festhalten, zunächst ein Beispiel.

Beispiel 3.2.1 (Satz von Euler)**.**

Eine der klassischen Anwendungen des Satzes von Euler ist die Vereinfachung von Potenzen in Modulgleichungen:

Es soll berechnet werden: $\mathsf{Rest}(2^{26}, 21)$. Hierzu verwenden wir $\varphi(21) = \varphi(3 \cdot 7) = 2 \cdot 6 = 12$ zusammen mit dem Satz von Euler: Es gilt $2^{12} \equiv 1 \pmod{21}$, da $\mathrm{ggT}(2, 21) = 1$ gilt.

Es folgt, dass $2^m \equiv 2^{m-12} \pmod{21}$ gilt:

$$\underbrace{2^{26}}_{\text{Gesamt: mod } 21} = 2^{14} \cdot 2^{12} \equiv \underbrace{2^{14}}_{\text{Exponent: mod } \varphi(21)} \pmod{21}$$

Dies wendet man mehrfach an und erhält: $2^{26} \equiv 4 \pmod{21}$.

$$2^{26} \overset{-\varphi(21)}{\equiv} 2^{14} \overset{-\varphi(21)}{\equiv} 2^{2} \pmod{21}$$

Insgesamt gilt also: $2^{26} \equiv 2^{\mathsf{Rest}(26,12)} \pmod{21}$.

Lemma 3.2.2. Für $a, n \in \mathbb{N}$ mit $n \geq 2$ und $\mathbf{ggT(a, n) = 1}$ gilt $a^m \equiv a^{\mathsf{Rest}(m,\varphi(n))} \pmod{n}$ für alle $m \in \mathbb{N}$

Beweis. Es seien $a, m, n \in \mathbb{N}$ und es gelte $\mathrm{ggT}(a, n) = 1$. Dann gilt nach Satz von Euler $a^{\varphi(n)} \equiv 1 \pmod{n}$.

Sei nun $r := \mathsf{Rest}(m, \varphi(n))$ dann gibt es ein $\ell \in \mathbb{N}$ mit $m = \ell \cdot \varphi(n) + r$. Daraus folgern wir

$$\begin{aligned} a^m &= a^{\ell\cdot\varphi(n)+r} = a^{\ell\cdot\varphi(n)} \cdot a^r \\ &= \underbrace{(a^{\varphi(n)})}_{\equiv 1 \pmod{n}}{}^{\ell} \cdot a^r \equiv 1^{\ell} \cdot a^r \pmod{n} \end{aligned}$$

□

Beispiel 3.2.3.

Es soll berechnet werden: $\mathsf{Rest}(5^{12014}, 21)$. Hierzu verwenden wir $\varphi(21) = \varphi(3 \cdot 7) = 2 \cdot 6 = 12$ zusammen mit dem Satz von Euler: Es gilt $5^{12} \equiv 1 \pmod{21}$, da $\mathrm{ggT}(5, 21) = 1$ gilt.

Man kann also (s. Lemma 3.2.2) im Exponenten 12014 alle Vielfachen von 12 entfernen. Es gilt $12014 = 12 \cdot 10^3 + 12 + 2$ bzw. $\mathsf{Rest}(12013, 12) = 2$.

Es folgt, dass $5^{12013} \equiv 5^2 \pmod{21}$ gilt, und damit gilt

$$\mathsf{Rest}(5^{12013}, 21) = \mathsf{Rest}(5^2, 21) = \mathsf{Rest}(25, 21) = 4.$$

3.2.2. Allgemeines schnelles Potenzieren

Will man nun eine Potenz a^m berechnen und ist nicht an dem Rest modulo einer Zahl n interessiert, führt man mit dem naiven Ansatz $m - 1$ Multiplikatonen durch.

$$a^m = \underbrace{a \cdot a \cdot \ldots \cdot a}_{(m-1)-\text{oft}}$$

Das lässt sich beschleunigen, wenn man sukzessive Potenzen mit Zweierpotenzen berechnet. Dazu berechnet man die Binärdarstellung von m. Setze dazu $s = \lfloor \log_2(m) \rfloor$ und beobachte, dass die Binärdarstellung den Koeffizienten der folgenden Darstellung entspricht:

$$m = \sum_{i=0}^{s} b_i \cdot 2^i \qquad \text{mit } b_i \in \{0, 1\}$$

Mit Hilfe der Binärdarstellung lässt sich die Potenz a^m geschickt umschreiben. Es ist nämlich

$$a^m = a^{\sum_{i=0}^{s} b_i \cdot 2^i} = a^{b_0 \cdot 2^0 + b_1 \cdot 2^1 + b_2 \cdot 2^2 + \ldots + b_s \cdot 2^s} = a^{b_0 \cdot 2^0} \cdot a^{b_1 \cdot 2^1} \cdot a^{b_2 \cdot 2^2} \cdot \ldots \cdot a^{b_s \cdot 2^s}$$

Es reicht also den folgenden Pool von Zahlen zu berechnen.

$$\mathcal{P} = \left\{ a, a^2, a^4, a^8, \ldots, a^{2^i}, \ldots, a^{2^s} \right\}$$

Dieser Pool $\mathcal{P}$ beinhaltet $s+1$ Zahlen und die folgende Beobachtung bringt die Beschleunigung: Die Zahlen in $\mathcal{P}$ kann man einfach sukzessive berechnen. Denn es ist

$$a^{2^i} = a^{2^{i-1}} \cdot a^{2^{i-1}}$$

$$\Rightarrow \quad \mathcal{P} = \left\{ a, a^2 = a \cdot a, a^4 = a^2 \cdot a^2, a^8 = a^4 \cdot a^4, \ldots, a^{2^i} = a^{2^{i-1}} \cdot a^{2^{i-1}}, \ldots, a^{2^s} = a^{2^{s-1}} \cdot a^{2^{s-1}} \right\}$$

Man benötigt also $s \sim \log_2(m)$ Multiplikationen um $\mathcal{P}$ zu berechnen.

In algorithmischer Schreibweise liest sich das Schema:

Algorithmus 3.2.4 (SPOT)**.**

Input: $a, m \in \mathbb{N}$.
Output: a^m.
1. Berechne die Binärdarstellung von $m = \sum_{i=0}^{s} b_i \cdot 2^i$ mit $b_i \in \{0, 1\}$.
2. Setze $\mathcal{P} = \{a\}$.
for $i = 1$ **to** $\lfloor \log_2(m) \rfloor$ **do**
4. Lese $a^{2^{i-1}}$ aus $\mathcal{P}$ aus und berechne $a^{2^i} = a^{2^{i-1}} \cdot a^{2^{i-1}}$.
5. Speichere a^{2^i} in $\mathcal{P}$.
end for
6. Lese alle a^{2^i} aus $\mathcal{P}$ aus, für welche $b_i = 1$.
7. Berechne

$$a^m = \prod_{0 \le i \le \lfloor \log_2(m) \rfloor : b_i = 1} a^{2^i}.$$

Wir fassen unsere in der Prosa schon ausgeführte Beobachtung über die Komplexitätsreduktion in dem folgenden Lemma zusammen.

Lemma 3.2.5. Seien $a, m \in \mathbb{N}$. Der Algorithmus SPOT benötigt höchstens $2 \cdot \lfloor \log_2(m) \rfloor$ Multiplikationen um a^m zu berechnen.

Beweis. Wir haben uns schon überlegt, dass $\lfloor \log_2(m) \rfloor$ viele Multiplikationen benötigt werden um $\mathcal{P}$ zu erzeugen und das mit Hilfe der Werte in $\mathcal{P}$ der Wert a^m mittels Schritt 7 berechnet werden kann. Schritt 7 benötigt höchstens $\lfloor \log_2(m) \rfloor$ weitere Multiplikationen. □

Beispiel 3.2.6.

Wähle $a = 7$ und $m = 10$. Es sind naiv 9 Multiplikationen nötig um $7^{10} =$ zu berechnen.

Binärdarstellung von 10: $10 = 2 + 8 = 0 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 1 \cdot 8$

3.3. Kryptographische Anwendung: Das RSA-Verfahren

In der Kryptographie geht es darum, Nachrichten sicher zu transportieren, d.h. eine Nachricht $\mathcal{A}$ so zu verändern, dass ein Dritter aus der veränderten Nachricht $\mathcal{C}$ (dem Chiffrat) nicht auf die Originalnachricht $\mathcal{A}$ schließen kann.

Beispiel 3.3.1 (One-Time-Pad)**.**

Ein einfaches und sicheres Verfahren in der Kryptographie ist das Verwenden sogenannter „One-Time-Pads“ oder „One-Time-Keys“ .

Ist ein Bit-String $\mathcal{A} := (b_1, \ldots, b_n) \in \{0,1\}^n$ der Länge n zu verschlüsseln, so generiert man einen zufälligen Bitstring $\mathcal{K} := (k_1, \ldots, k_n) \in \{0,1\}^n$ (der one-time-*key*) und addiert die passenden Elemente modulo 2, d.h. $\mathcal{C} := (b_1 \oplus_2 k_1, \ldots, b_n \oplus_2 k_n)$

Fasst man die Bitstrings als Vektoren über $\mathbb{Z}_2$ auf, und $\oplus$ als die elementweise Addion modulo 2, so gilt

$$\mathcal{C} = \mathcal{A} \oplus \mathcal{K} \qquad \text{und} \qquad C \oplus \mathcal{K} = \mathcal{A} \oplus (\mathcal{K} \oplus \mathcal{K}) = \mathcal{A} \oplus (0, \ldots, 0) = \mathcal{A}.$$

Das Entschlüsseln der Verschlüsselten Nachricht erfolgt also (genau wie das Verschlüsseln) durch Addtion von $\mathcal{K}$.

Sicherheit: Gelingt es, die Original-Nachricht $\mathcal{A}$ aus $\mathcal{C}$ zu ermitteln, so kennt man $\mathcal{A}$ und $\mathcal{C}$, und damit auch $\mathcal{K}$, denn es gilt:

$$\mathcal{A} \oplus \mathcal{C} = \overbrace{\mathcal{A} \oplus \mathcal{A}}^{(0,\ldots,0)} \oplus \mathcal{K} = \mathcal{K}$$

Dies bedeutet, dass das Verfahren im folgenden Sinne sicher ist: Ein Unbefugter hat genau dann eine Chance, eine Nachricht zu dechiffrieren, wenn er $\mathcal{K}$ bestimmen kann. Ist $\mathcal{K}$ „sicher“ und unbestimmbar, so kann auch $\mathcal{A}$ nicht aus $\mathcal{C}$ ermittelt werden.

Der Nachteil von One-Time-Pads ist, dass das Pad $\mathcal{K}$ beiden Kommunikationsparteien bekannt sein muss. Die Kommunikationsparteien müssen „sich einmal vorab treffen“ bzw. genauer gesagt, die Kommunikationsparteien müssen $\mathcal{K}$ vorab einmal über einen abhörsicheren Kanal austauschen.

Das Problem: Der Austausch von $\mathcal{K}$ kann nicht (wirklich) mittels eines (anderen) One-Time-Pads $\widetilde{\mathcal{K}}$ erfolgen, denn dazu müsste zunächst $\widetilde{\mathcal{K}}$ ausgetauscht werden, etc. etc.

Es muss also auf andere Weise ein sicherer Kommunikationskanal erzeugt werden. Genau dies leisten sogenannte *Public-Key*-Verfahren.

3.3.1. Das RSA-Schema.

Für die klassischen Chiffrierverfahren besteht ein Sicherheitsproblem darin, dass man den Schlüssel $\mathcal{K}_e$ der Codiervorschrift (ebenso wie die Schlüssel $\mathcal{K}_d$ der Decodiervorschrift) geheimhalten und zuvor vereinbaren muss (beim One-time-pad also $\mathcal{K}_e = \mathcal{K}$). Dies erzeugt ein „Henne-Ei"-Problem, denn der Schlüssel $\mathcal{K}_e$ kann nicht ohne Verschlüsselung sicher zwischen den Parteien kommuniziert werden.

Diffie und Hellman haben 1976 den (damals völlig neuartigen) Vorschlag der sogenannten Public-key-Kryptographie gemacht:

Jeder Teilnehmer des Systems besitzt

- einen ***öffentlichen Schlüssel*** e (zum ***E***ncodieren) und
- einen ***geheimen Schlüssel*** d (zum ***D***ecodieren).

Die Ver- und Entschlüsselalgorithmen E (encode) und D (decode) müssen nun die spezielle Eigenschaft haben, dass

- Für $\mathcal{C} := E(\mathcal{A})$ stets $D(\mathcal{C}) = \mathcal{A}$ gilt (D ist Umkehrfunktion zu E).
- $E(\mathcal{A})$ für eine Nachricht $\mathcal{A}$ mittels e leicht berechnet werden kann,
- $D(\mathcal{C})$ mit Kenntnis von d leicht zu berechnen ist,
- $D(\mathcal{C})$ ohne Kenntnis von d jedoch „sehr schwer" zu berechnen ist. („*Trapdoor-Funktion*")

Das bekannteste öffentliche Chiffriersystem ist das 1978 von Rivest, Shamir und Adleman vorgeschlagene RSA-Schema. Es beruht darauf, dass es schwer ist, eine Zahl m in ihre Primfaktoren zu zerlegen. In Folgenden Schema ist das Verfahren erklärt.

RSA-Verfahren

► Schlüsselerzeugung:

- Wähle (sehr große) Primzahlen p, q und berechne $\mathsf{N} := p \cdot q$
  - Berechne $\varphi(\mathsf{N}) := (p-1) \cdot (q-1)$
  - Wähle $\mathsf{e} \in \mathbb{Z}^*_{\varphi(\mathsf{N})}$ d.h. e und $\varphi(\mathsf{N})$ sind teilerfremd.
  - Berechne $\mathsf{d} := \mathsf{e}^{-1}$ in $\mathbb{Z}^*_{\varphi(\mathsf{N})}$

    Bestimme s, t mittels erweitertem euklidischem Algorithmus, so dass gilt $s \cdot \mathsf{e} + t \cdot \varphi(\mathsf{N}) = 1$. Setze $\mathsf{d} = \mathsf{Rest}(s, \varphi(\mathsf{N}))$
  - Lösche p, q und $\varphi(\mathsf{N})$
- Veröffentliche: e, N.
- Halte geheim: d

► Encodierung:

- Nachricht: $a \in \{2, \ldots, \mathsf{N}-1\}$
- Berechne: $c := \mathsf{Rest}(a^{\mathsf{e}}, \mathsf{N})$

► Decodierung:

▶ Berechne: $a = \mathsf{Rest}(c^{\mathsf{d}}, \mathsf{N})$

Zur Veranschaulichung folgt direkt ein Beispiel.

Beispiel 3.3.2.

▶ **Schlüsselerzeugung:** Sei $p = 5$, $q = 11$. Dann ist

$$\begin{aligned} \mathsf{N} &= 5 \cdot 11 = 55 \\ \varphi(\mathsf{N}) &= 4 \cdot 10 = 40 \end{aligned}$$

Der Encode-Exponent e muss teilerfremd zu $40 = 2^3 \cdot 5$ sein, darf also nicht durch 2 oder 5 teilbar sein.

Wird als Verschlüsselungs-Exponent $\mathsf{e} := 3$ gewählt, dann ist $\mathsf{d} = 27$ (da $e \cdot d = 81 \equiv 1 \pmod{40}$).

▶ **Encodierung:** Für die Nachricht $a := \mathbf{4}$ berechnen wir das Chiffrat $\mathsf{Rest}(a^{\mathsf{e}}, \mathsf{N})$

$$c = \mathsf{Rest}(\mathbf{4}^3, 55) = \mathsf{Rest}(64, 55) = \mathbf{9}$$

▶ **Decodierung:** Das Chiffrat $c = \mathbf{9}$ liefert wieder $a = \mathbf{4}$, denn es gilt $\mathsf{Rest}(c^{\mathsf{d}}, \mathsf{N}) = \mathsf{Rest}(9^{27}, 55) = \mathbf{4}$

$$\begin{aligned} \mathbf{9}^{27} &= (9^3)^9 \equiv (14)^9 \pmod{55} && \mid 9^3 = 729 = 14 + 660 + 55 \\ 14^9 &= (14^3)^3 \equiv (-6)^3 \pmod{55} && \mid 14^3 = 2744 = -6 + 2200 + 550 \\ (-6)^3 &= -216 \equiv \mathbf{4} \pmod{55} && \mid -216 = 4 - 220 \end{aligned}$$

Wir möchten im Folgenden nachweisen, dass das RSA-Verfahren auch "funktioniert".

Lemma 3.3.3 (Beweis der Korrektheit)**.** Beim RSA-Schema gilt: Ist $c := \mathsf{Rest}(a^{\mathsf{e}}, \mathsf{N})$ so gilt $a = \mathsf{Rest}(c^{\mathsf{d}}, \mathsf{N})$.

Beweis. Zu zeigen ist hier, dass $c^{\mathsf{d}} \equiv a \pmod{\mathsf{N}}$ gilt, denn dann folgt aus $2 \leq a \leq \mathsf{N} - 1$ die Behauptung.

Nach Konstruktion von d gilt

$$\mathsf{e} \odot_{\varphi(\mathsf{N})} \mathsf{d} = \mathsf{Rest}\left(\mathsf{e} \cdot \mathsf{d}, \varphi(\mathsf{N})\right) = \mathbf{1}$$

es gilt also

$$\mathsf{e} \cdot \mathsf{d} = \mathbf{1} + k \cdot \varphi(\mathsf{N}) \qquad \text{für ein } k \in \mathbb{Z}.$$

Es gilt nun:

$$\begin{aligned} c^{\mathsf{d}} = \mathsf{Rest}(a^{\mathsf{e}}, \mathsf{N})^{\mathsf{d}} \equiv \overbrace{(a^{\mathsf{e}})^{\mathsf{d}}}^{=a^{\mathsf{e}\cdot\mathsf{d}}} &\equiv a^{\overbrace{k\cdot\varphi(\mathsf{N})+\mathbf{1}}^{\mathsf{e}\cdot\mathsf{d}}} \pmod{\mathsf{N}} \\ &\equiv a^{k\cdot\varphi(\mathsf{N})} \cdot a^{\mathbf{1}} \pmod{\mathsf{N}} \\ &\overset{(\star)}{\equiv} a \pmod{\mathsf{N}} \end{aligned}$$

Für $\mathrm{ggT}(a, \mathsf{N}) = 1$ folgt die Gleichung $(\star)$ aus dem Satz von Euler, da dann $a^{\varphi(\mathsf{N})} \equiv 1 \pmod{\mathsf{N}}$ gilt.

Um die Gleichung $(\star)$ im allgemeinen Fall zu zeigen verwendet man den chinesischen Restsatz. Wir beweisen nun $(\star)$ komplett, dh. der obige Fall $\mathrm{ggT}(a, \mathsf{N}) = 1$ ist im Folgenden „inklusive“.

Wir zeigen zunächst, dass gilt:

$$\left.\begin{array}{rcll} a^{\mathsf{e}\cdot\mathsf{d}} & \equiv & a & (\mathrm{mod}\ p) \\ a^{\mathsf{e}\cdot\mathsf{d}} & \equiv & a & (\mathrm{mod}\ q) \end{array}\right\} (\star\star)$$

1. Annahme: es gelte $p|a$. Dann gelten $a \equiv 0 \pmod p$ und $a^{e\cdot d} \equiv 0 \pmod p$.
   Aus der Transitivität der Modulrechnung folgert man $a^{\mathsf{e}\cdot\mathsf{d}} \equiv a \pmod p$.
2. Annahme: es gelte $p \nmid a$. Es gilt gilt $a^{\varphi(\mathsf{N})} \equiv 1 \pmod p$, denn:
   Die Zahlen a und p sind teilerfremd: Teiler von p sind 1 und p. Weiter gilt $p \nmid a$, d.h. $\mathrm{ggT}(a, p) = 1$.
   Nach Satz von Euler gilt mit $\varphi(p) = p - 1$ die Gleichung $a^{p-1} \equiv 1 \pmod p$ und damit

   $$a^{\varphi(\mathsf{N})} = a^{(p-1)(q-1)} = \left(a^{p-1}\right)^{q-1} \equiv (1)^{q-1} \pmod p$$

   Es gilt also $a^{\mathsf{e}\cdot\mathsf{d}} \equiv a \pmod p$ wegen

   $$a^{\mathsf{e}\cdot\mathsf{d}} = a^{k\cdot\varphi(\mathsf{N})+1} = \left(a^{\varphi(\mathsf{N})}\right)^k \cdot a^1 \equiv (1)^k \cdot a \pmod p$$

3. Die Aussage $a^{\mathsf{e}\cdot\mathsf{d}} \equiv a \pmod q$ für q zeigt man analog zur Fallunterscheidung für p.

Betrachtet man $(\star\star)$, so sieht man, dass sowohl $\widetilde{x} := a^{\mathsf{e}\cdot\mathsf{d}}$ als auch $\widehat{x} := a$ jeweils Lösungen sind von

$$\left.\begin{array}{rcll} x & \equiv & a & (\mathrm{mod}\ p) \\ x & \equiv & a & (\mathrm{mod}\ q) \end{array}\right\} (\star\star\star)$$

Die Moduln p, q sind teilerfremd. Nach Chinesische Restsatz sind also die Lösungen von $(\star\star\star)$ äquivalent modulo $\mathsf{N} = p \cdot q$, es gilt $\widetilde{x} = \widehat{x} + k \cdot \mathsf{N}$ mit $k \in \mathbb{Z}$. Dies liefert $a^{\mathsf{e}\cdot\mathsf{d}} \equiv a \pmod{\mathsf{N}}$. □

Bemerkung 3.3.4 (Krypt-Analyse)**.**

Das Lemma 3.3.3 beweist, dass das RSA-Verfahren in so fern funktioniert, als dass aus dem erzeugten Chiffrat der ursprüngliche Message eindeutig ermittelt werden kann (in der Kenntnis des geheimen Schlüssels d). Ob das Verfahren aber auch “sicher” ist, muss gesondert diskutiert werden. Hierbei wird “sicher” ganz vielfältig interpretiert. Sicher in Bezug auf das entschlüsseln eines abgehörten Chiffrats oder das Auffinden des geheimen Schlüssels durch verschiedene Angriffe.

In Bezug auf das Berechnen des geheimen Schlüssels d aus dem öffentlichen Schlüssel e ist das RSA-Verfahren sicher, als dass man dazu $\varphi(\mathsf{N})$ benötigt.

Die Kenntnis von $\varphi(\mathsf{N}) = (p-1)(q-1)$ ist jedoch äquivalent zur Kenntnis von p und q, d.h. der Faktorisierung von N. Man kann $p + q$ und $p - q$ mit leicht durführbaren Operationen aus N und $\varphi(\mathsf{N})$ berechnen (und erhält so p und q):

$$\left.\begin{array}{rcl} \varphi(\mathsf{N}) = (p-1)(q-1) & = & pq - (p+q) + 1 \\ & = & \mathsf{N} - (p+q) + 1 \end{array}\right\} \quad \Rightarrow \quad (p+q) = \mathsf{N} + 1 - \varphi(\mathsf{N})$$

$$\left.\begin{array}{rcl} (p-q)^2 & = & p^2 - 2pq + q^2 \\ & = & p^2 + 2pq + q^2 - 4pq \\ & = & (p+q)^2 - 4\mathsf{N} \end{array}\right\} \quad \Rightarrow \quad (p-q) = \sqrt{(p+q)^2 - 4\mathsf{N}}$$

Wer also aus N die Zahl $\varphi(\mathsf{N})$ effizient (bzw. schnell) berechnen kann, der kann N auch schnell faktorisieren. Für das Faktorisieren von Zahlen (z.B. von N) ist aber bisher kein effizienter („schneller“) Algorithmus bekannt.

Dieses Argument ist beispielhaft für die Reduktion in der Komplexitätstheorie. Man reduziert das Berechnen des geheimen Schlüssels aus dem öffentlichen Schlüssel (und N) im RSA-Verfahren auf das Faktorisieren großer Zahlen. Ist das Berechnen des geheimen Schlüssels einfach, so auch das Faktorisieren. Das Berechnen des geheimen Schlüssels im RSA-Verfahrens ist einfacher als das Faktorisieren großer Zahlen. Ist man in der Lage den Schlüssel (effizient) zu berechnen, so auch die Faktorisierung großer Zahlen. Das RSA-Verfahren ist also sicher in Bezug auf das Extrahieren des geheimen Schlüssels aus dem öffentlichen Schlüssel, so lange das Faktorisierung als schwer angenommen wird. Wird das Faktorisieren einfach (also ein effizientes Verfahren gefunden) heißt das im Umkehrschluss aber noch nicht, dass das RSA-Verfahren unsicher ist.

Exkurs 3.3.5 (Faktorisierungs-Challenge)**.**

Um ein Gefühl für die Schwierigkeit des Faktorisierens großer Zahlen zu vermitteln, dienen die folgenden Anhaltspunkte. Die RSA Laboratories haben regelmäßig Zahlen der Form $N = p \cdot q$ als Faktorisierungs-Challenge veröffentlicht:

- Im Jahr 1977 wurde die 129-stellige Dezimalzahl „RSA-129“ als Herausforderung für das RSA-System veröffentlicht (Preisgeld von $100). Diese Zahl wurde erst 1994 faktorisiert, unter Beteiligung von 600 Freiwilligen und einem Rechenaufwand von ca. 5000 MIPS-Jahren.
- Im Jahr 2003 wurde mit 5-monatiger Rechenleistung auf 120 Maschinen die Zahl RSA-576 der Bitlänge 576 faktorisiert (Preisgeld $10.000).
- Im Jahr 2005 wurde die Zahl RSA-640 der Bitlänge 640 faktorisiert (Preisgeld $20.000).
- Das letzte Preisgeld betrug $100.000 auf eine Zahl RSA-1024 (1024-Bit, 309 Dezimalstellen) und $200.000 auf eine Zahl RSA-2048 (2048 Bit, 617 Dezimalstellen).

Obwohl mittlerweile für das Faktorisieren der RSA-Challenge-Zahlen keine Prämien mehr gezahlt werden, wurde im Dezember 2009 die Zahl RSA-768 faktorisiert.

Nicht vorhersehbare Entwicklungen, wie die Entwicklung deutlich schnellerer Algorithmen oder eines Quantencomputers, der die Faktorisierung von Zahlen durch Verwendung des Shor-Algorithmus effizient durchführen könnte, bergen zumindest für die mittel- und langfristige Sicherheit der RSA-verschlüsselten Daten gewisse Risiken.

3.3.2. Angriffe gegen das unmodifizierte RSA-Verfahren

In der Praxis wird das RSA-Verfahren in der oben beschrieben Form nicht eingesetzt - es hat in dieser Form mehrere Schwächen. Eine Schwäche entsteht durch zu kleine Klartexte a, wie sie z.B. im Chat auftreten.

Wenn der Klartext a und der Verschlüsselungsexponent e beide klein sind, so dass $a^e < N$ gilt, so folgt $c = \mathsf{Rest}(a^e, N) = a^e$. In diesem Fall kann ein Angreifer die e-te Wurzel aus c berechnen und das Chiffrat c auf diese Weise entschlüsseln, denn:

Gewöhnliches Wurzelziehen (z.B. $\sqrt[e]{c}$) aus ganzen Zahlen ist eine leicht zu bewerkstelligende Rechenoperation, nur das Wurzelziehen modulo einer großen Zahl N *ist schwierig.*

Um solche Angriffe zu verhindern, wird ein *Padding-Verfahren* eingesetzt:

Man hängt der Bitfolge $(a)_2$ des Klartextes a eine zufällige Bitfolge R an. R hat eine vorgegebene Struktur, die unter mehreren möglichen zufällig gewählt wird (s. z.B. ISO 9796). Verschlüsselt wird nun die als Zahl aufgefasste Bitfolge $\widetilde{a} := (a, R)_2$, wobei nun gilt $(\widetilde{a})^e > N$ und damit $c = \mathsf{Rest}((\widetilde{a})^e, N) \neq (\widetilde{a})^e$.

Beispiel 3.3.6.

Sei $p = 7$, $q = 11$. Dann ist

$$\begin{array}{rcl} N & = & 7 \cdot 11 \;=\; 77 \\ \varphi(N) & = & 6 \cdot 10 \;=\; 60 \end{array}$$

Als Verschlüsselungs-Exponent kommen nur Zahlen in Frage, die teilerfremd zu $60 = 2^2 \cdot 3 \cdot 5$ sind, die also nicht durch 2 oder 3 oder 5 teilbar sind.

Wird als Verschlüsselungs-Exponent $e := 13$ gewählt, dann ist $d = 37$ (da $d \cdot e = 381 \equiv 1 \pmod{\varphi(N)}$.

Wir verschlüsseln nun die Nachricht $a := 2$, d.h. wir berechnen $\mathsf{Rest}(2^e, N)$.

Das Berechnen von a^e kann man durch sukzessives Quadrieren beschleunigen: Aus der Binärdarstellung von $e = 13 = 8 + 4 + 1$ entnimmt man $a^e = a^8 \cdot a^4 \cdot a^1$ und berechnet durch Quadrieren (modulo N) sukzessive die Reste von a^2, $a^4 = \left(a^2\right)^2$, $a^8 = \left(a^4\right)^2$. Es gilt

$$\begin{array}{rcl} 2^2 & = & 4 \qquad\qquad\quad \equiv 4 \pmod{77} \\ 2^4 & = & (4)^2 \;=\; 16 \;\equiv 16 \pmod{77} \\ 2^8 & = & (16)^2 \;=\; 256 \;\equiv 25 \pmod{77} \end{array}$$

Und damit ergibt sich: $\mathsf{Rest}(a^e, N) = \mathsf{Rest}\,(25 \cdot 16 \cdot 2,\ \ 77) = \mathsf{Rest}(800, 77) = 30$.

Entschlüsselt man die Zahl 30 wieder auf dem selben Weg, berechnet man zunächst

$$\begin{array}{rcl} 30^2 & = & 900 \qquad\quad \equiv 53 \pmod{77} \\ 30^4 & \equiv & 53^2 \;\equiv\; 2809 \;\equiv 37 \pmod{77} \\ 30^8 & \equiv & 37^2 \;\equiv\; 1369 \;\equiv 60 \pmod{77} \\ 30^{16} & \equiv & 60^2 \;\equiv\; 3600 \;\equiv 58 \pmod{77} \\ 30^{32} & \equiv & 58^2 \;\equiv\; 3364 \;\equiv 53 \pmod{77} \end{array}$$

Beim Entschlüsseln von $c = 30$ entsteht also wieder $a = 2$, denn es gilt $\mathsf{Rest}(c^d, N) = 2$ wegen

$$\begin{array}{rcl} 30^{37} & = & 30^{32} \cdot 30^4 \cdot 30^1 \\ & \equiv & 53 \quad \cdot 37 \quad \cdot 30 \equiv 764 \cdot 77 + 2 \equiv 2 \pmod{77} \end{array}$$

3.3.3. RSA-Signaturschema

Mit Hilfe eines Signaturverfahrens lässt sich die Authentizität der Verfasserschaft einer Nachricht überprüfen oder belegen. Der Sender versieht eine Nachricht mit einer Signatur, die ihn eindeutig als ihr Verfasser ausweist, er

unterschreibt die Nachricht sprichwörtlich. Ein “Fälscher” ist nicht in der Lage diese Signatur nachzuahmen/zu fälschen.

Das Signaturverfahren mittels RSA fußt darauf, dass das Ver- und Entschlüsseln im RSA-Verfahren inverse Abbildungen sind. Ist (e, N) das öffentliche Schlüsselpaar und d der private Schlüssel, so gilt

- das Verschlüsseln einer Nachricht x liefert $E(x) := \mathsf{Rest}(x^{\mathsf{e}}, \mathsf{N})$,
- das Entschlüsseln eines Chiffrretextes z liefert $D(z) := \mathsf{Rest}(z^{\mathsf{d}}, \mathsf{N})$.

Die Funktion $D(\cdot)$ ist die Inverse zu $E(\,\cdot\,)$. Das heißt:

- ▶ Es gilt $D(\,E(x)\,) = x$. (Entschlüsseln des Chiffrats $E(x)$ liefert x).
- ▶ Es gilt aber auch umgekehrt $E(D(z)) = z$, denn:

$$\begin{aligned}
&\text{a)} \quad E(x) &&= \mathsf{Rest}\,(\,x^{\mathsf{e}}\,, \mathsf{N}) &&\equiv x^{\mathsf{e}} &&\pmod{\mathsf{N}}\\
&\text{b)} \quad E(\,D(z)\,) &&= \mathsf{Rest}\,(\,D(z)^{\mathsf{e}}\,, \mathsf{N}) &&\equiv D(z)^{blue\,\mathsf{e}} &&\pmod{\mathsf{N}}\\
& &&&&\overset{\text{a)}}{\equiv} z^{\mathsf{d}\cdot\mathsf{e}} &&\pmod{\mathsf{N}}
\end{aligned}$$

Wir haben in der Krypt-Analyse des RSA-Verfahrens gezeigt, dass es *in der Praxis unmöglich ist* d aus den Zahlen e, N zu berechnen. „Praktisch unmöglich“ heißt hier, der Rechenaufwand ist größer als die momentan verfügbaren Rechner ermöglichen.

Dies bedeutet aber auch, dass jemand, der d nicht kennt, für x nicht $D(x) := \mathsf{Rest}(x^{\mathsf{d}}, \mathsf{N})$ berechnen kann. Nur derjenige, der die Schlüssel e, N erzeugt hat, kennt d und kann für x ein Paar $(x, D(x))$ berechnen.

Für ein Paar (x, z) lässt sich jedoch mit e, N leicht püfen, ob $z = D(x)$ gilt:

Ist $z = \mathsf{Rest}(x^{\mathsf{d}}, \mathsf{N})$ so muss $\mathsf{Rest}(z^{\mathsf{e}}, \mathsf{N}) = x$ gelten, wegen

$$\mathsf{Rest}(z^{\mathsf{e}}, \mathsf{N}) = \mathsf{Rest}((x^{\mathsf{d}})^{\mathsf{e}}, \mathsf{N}) = \mathsf{Rest}(x^{\mathsf{d}\cdot\mathsf{e}}, \mathsf{N}) = x.$$

Bemerkung 3.3.7 (Signaturschema mit RSA)**.**

Person A (Alice) möchte eine Nachricht a an Person B (Bob) schicken und signieren.

A besitzt den privaten Schlüssel d und hat das Paar (e, N) veröffentlicht.

Wir gehen davon aus, dass B sicher sein kann, dass das Paar (e, N) tatsächlich von A stammt.

1. Person A verschlüsselt a mit dem *eigenen privaten Schlüssel* d, berechnet also $c := \mathsf{Rest}(a^{\mathsf{d}}, \mathsf{N})$
2. Person A sendet (a, c) an Person B.
3. Person B erhält ein Paar (x, y). Er prüft nun, ob (x, y) von A stammt:
   - ▶ Er entschlüsselt y mit A’s öffentlichem Schlüssel e, d.h. er berechnet $z := \mathsf{Rest}(y^{\mathsf{e}}, \mathsf{N})$.
   - ▶ Gilt $x = z$ so stammt (x, y) tatsächlich von A, denn nur A ist in der Lage aus x das passende $y = \mathsf{Rest}(x^{\mathsf{d}}, \mathsf{N})$ zu berechnen.

Der Schritt 2 sollte verschlüsselt erfolgen:

Eigentlich verschlüsselt A die (besondere) Nachricht $\overline{a} := (a, c)$ mit B's öffentlichen Schlüsseln $(\mathtt{e}_B, \mathtt{N}_B)$ und erhält $\overline{c} = \mathsf{Rest}((\overline{a})^{\mathtt{d}_B}, \mathtt{N}_B)$.

Der Empfänger B entschlüsselt dann $\overline{c}$ mit seinem privaten Schlüssel $\mathtt{d}_B$, erhält ein paar (x, y) und untersucht dies wie im dritten Schritt beschrieben.

Teil II.

Numerik

A Anhang

A.1. Der euklidische Algorithmus in Tabellenform

Der Eukidischen Algorithmus in Form von Algorithmus 2.1.3 lässt sich angenehm übersichtlich in Tabellenform durchführen.

Init

Laufindex j	Faktor Vor-Vor-r / Vor-r m_j	aktueller Rest r_j	Faktor vor a s_j	Faktor vor b t_j
0	/	a	1	0
1		b	0	1

Berechnen von m_1 in Zeile 1

Laufindex j	Faktor Vor-Vor-r / Vor-r m_j	aktueller Rest r_j	Faktor vor a s_j	Faktor vor b t_j
0	/	a	1	0
1	$\lfloor \frac{a}{b} \rfloor$	b	0	1

Berechnen von r_2, s_2, t_2 in Zeile 2

Laufindex j	Faktor Vor-Vor-r / Vor-r m_j	aktueller Rest r_j	Faktor vor a s_j	Faktor vor b t_j
0	/	a	1	0
1	$\lfloor \frac{a}{b} \rfloor$	b	0	1
2		r_2	1	$-\lfloor \frac{a}{b} \rfloor$

a	1	0
$-\lfloor \frac{a}{b} \rfloor \cdot b$	$-\lfloor \frac{a}{b} \rfloor \cdot 0$	$-\lfloor \frac{a}{b} \rfloor \cdot 1$
$= r_2$	$= 1$	$-\lfloor \frac{a}{b} \rfloor$

$r_2 = a - \lfloor \frac{a}{b} \rfloor \cdot b$

$s_2 = 1 - \lfloor \frac{a}{b} \rfloor \cdot 0 = 1$

$t_2 = 0 - \lfloor \frac{a}{b} \rfloor \cdot 1 = -\lfloor \frac{a}{b} \rfloor$

Berechnen von m_2 in Zeile 2

Laufindex j	Faktor Vor-Vor-r / Vor-r m_j	aktueller Rest r_j	Faktor vor a s_j	Faktor vor b t_j
0	/	a	1	0
1	$\lfloor \frac{a}{b} \rfloor$	b	0	1
2	$\lfloor \frac{b}{r_2} \rfloor$	r_2	1	$-\lfloor \frac{a}{b} \rfloor$

...