Vorlesung „Kryptographie“ – Dietzfelbinger

4 Zahlentheorie und Algorithmen

Zu Aussagen, die mit $(*)$ markiert sind, gibt es Beweise oder Anmerkungen im Anhang A. Beweise von rein zahlentheoretischen Aussagen sind nicht prüfungsrelevant. Beweise für Wahrscheinlichkeitsaussagen und Begründungen für Rechenzeiten von Algorithmen dagegen sind prüfungsrelevant.

4.1 Fakten aus der Zahlentheorie und grundlegende Algorithmen

Unsere Zahlenbereiche:

- $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$,
- $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$

Wir stellen uns die Zahlen immer als zu einer passenden Basis b dargestellt vor: Binärdarstellung, (Oktaldarstellung,) Dezimaldarstellung, Hexadezimaldarstellung, Darstellung zur Basis 256 (eine Ziffer ist ein Byte) oder 2^{32} oder 2^{64} (eine Ziffer ist ein 32- bzw. 64-Bit-Wort, passend für die Darstellung in einem Rechner).

Die Anzahl der Ziffern in der Darstellung von $a \in \mathbb{N}$ zur Basis b ist $\lceil \log_b(a+1) \rceil$. Das ist etwa $\frac{\ln a}{\ln b} = \frac{\log a}{\log b}$.

Verwendete Operationen: Addition, Subtraktion, Multiplikation, Division mit Rest.

Wir nehmen an, dass zwei einziffrige Zahlen in Zeit $O(1)$ addiert und subtrahiert werden können („von der Hardware“). Addition zweier n-ziffriger Zahlen kostet dann Zeit $O(n)$, Multiplikation einer n-ziffrigen und einer ℓ-ziffrigen Zahl mit der Schulmethode kostet Zeit $O(n\ell)$. Es gibt schnellere Verfahren: Karatsuba mit $O(n^{1{,}59})$ für zwei n-ziffrige Zahlen, Schönhage-Strassen (1977) sogar mit $O(n \log n \log\log n)$. Nach längerer Pause erschienen 2007 und 2008 Verbesserungen Verbesserungen. Im März 2019 erschien eine Arbeit[1], die zeigt, wie man zwei n-Bit-Zahlen in Zeit $O(n \log n)$ multiplizieren kann. Man vermutet, dass das optimal ist. (Nach aktuellem Stand ergeben sich Vorteile gegenüber Karatsuba aber erst für unrealistisch lange Zahlen.)

[1] David Harvey, Joris van der Hoeven. Integer multiplication in time $O(n \log n)$. Annals of Mathematics, Princeton University, Department of Mathematics, in press. `https://hal.archives-ouvertes.fr/hal-02070778v2/document`

Fakt 4.1 ***Division mit Rest***
Zu $x \in \mathbb{Z}$ und $m \geq 1$ gibt es ein r mit $0 \leq r < m$ und ein q mit $x = qm + r$. Die Zahlen q und r sind eindeutig bestimmt.

Die Zahl r („Rest“) bezeichnen wir mit $\boldsymbol{x}$ **mod** $\boldsymbol{m}$. Sie hat die Darstellung $x - qm$, unterscheidet sich also von x um ein Vielfaches von m. Der Quotient q wird mit $\boldsymbol{x}$ **div** $\boldsymbol{m}$ bezeichnet.[2]

Beispiel: $30 = 3 \cdot 9 + 3$, $30 \bmod 9 = 3$. $-30 = (-4) \cdot 9 + 6$, $(-30) \bmod 9 = 6$.

Aufwand für Division mit Rest: Die Division einer n-ziffrigen Zahl durch eine ℓ-ziffrige Zahl mit der Schulmethode kostet Zeit $O(n\ell)$.

Wir sagen, dass eine ganze Zahl y die ganze Zahl x *teilt* (oder dass y *ein Teiler von* x *ist*), wenn $x = qy$ für eine ganze Zahl q gilt. Oft schreibt man dafür kurz $y \mid x$. Wenn y kein Teiler von x ist, schreiben wir $y \nmid x$.

Beispiel: $3 \mid 12$, $-3 \mid 12$, $-3 \mid -12$, $-3 \mid 12$, $3 \mid 0$, $0 \mid 0$.

Beobachtungen: Die Teilbarkeitsrelation $\mid$ ist *reflexiv* und *transitiv*. Es handelt sich damit um eine „Präordnung“ (oft auch „Quasiordnung“ genannt). Zahlen x und $-x$ können von ihr nicht unterschieden werden: Es gilt $x \mid y \Leftrightarrow -x \mid y$ und $y \mid x \Leftrightarrow y \mid -x$, und weiter $x \mid -x$ *und* $-x \mid x$. Die Präordnung ist also nicht antisymmetrisch. Sie ist auch nicht total, weil manche Elemente nicht verglichen werden können: $4 \nmid 9$ und $9 \nmid 4$. Aus $0 \mid x$ folgt $x = 0$; für jede ganze Zahl y gilt $y \mid 0$; also ist in dieser Präordnung 0 das eindeutig bestimmte größte Element. Für jede ganze Zahl x gilt: $1 \mid x$ und $-1 \mid x$, also sind 1 und -1 kleinste Elemente (s. Abb. 1). Wenn $m \geq 1$ ist, ist $m \mid x$ gleichbedeutend mit $x \bmod m = 0$.

Fakt 4.2 ***Teilbarkeit***

Für beliebige $x, y, z \in \mathbb{Z}$ gilt:

(a) Aus $x \mid y$ und $x \mid z$ folgt $x \mid uy + vz$ für alle $u, v \in \mathbb{Z}$.

(b) Aus $x \mid y$ folgt $ux \mid uy$ für alle $u \in \mathbb{Z}$.

(c) Aus $x \mid y$ und $y \mid z$ folgt $x \mid z$ (Transitivität).

(d) Aus $x \mid y$ und $y \neq 0$ folgt $0 < |x| \leq |y|$.

(e) Aus $x \mid y$ und $y \mid x$ folgt $|x| = |y|$. Wenn zudem $x, y \geq 0$ gilt, folgt $x = y$.

[2] Mit der Notation $\lfloor \alpha \rfloor = \max\{k \in \mathbb{Z} \mid k \leq \alpha\}$ für beliebige reelle Zahlen α gilt x **div** $m = \lfloor x/m \rfloor$.

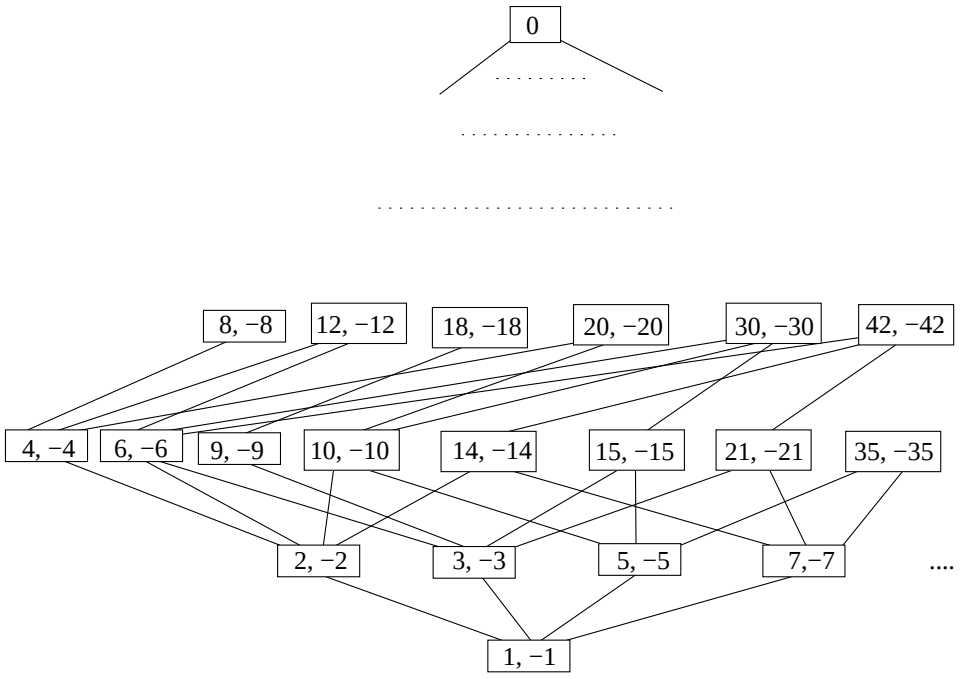

Abbildung 1: Einige Zahlen und ihre Teilbarkeitsbeziehungen. Beziehungen, die aus der Transitivität folgen, sind nicht eingetragen. Man erkennt 1 und -1 als kleinste Elemente und 0 als größtes Element der Teilbarkeitsbeziehung als Präordnung. Die Elemente in der Ebene unmittelbar über $\{1, -1\}$ sind die Primzahlen, positiv und negativ, also Zahlen $x \neq \pm 1$, die durch keine Zahl außer $\pm x$ und ± 1 teilbar sind.

Der *Beweis* ist eine einfache Übung.

Definition 4.3 ***Größter gemeinsamer Teiler***

(a) Für $x, y \in \mathbb{Z}$ heißt $t \in \mathbb{Z}$ ein **gemeinsamer Teiler** von x und y, wenn $t \mid x$ und $t \mid y$ gilt.
(*Bemerkung*: 1 ist stets gemeinsamer Teiler von x und y.)

(b) Für $x, y \in \mathbb{Z}$ sei $\mathbf{ggT}(\boldsymbol{x}, \boldsymbol{y})$, der **größte gemeinsame Teiler** von x und y, die (eindeutig bestimmte) nichtnegative Zahl d mit:
(i) d ist gemeinsamer Teiler von x und y;
(ii) jeder gemeinsame Teiler von x und y ist Teiler von d.

(c) $x, y \in \mathbb{Z}$ heißen **teilerfremd**, wenn $\mathrm{ggT}(x, y) = 1$ gilt, d. h. wenn sie nicht beide 0 sind und keine Zahl > 1 beide teilt.

Bei Definition 4.3(b) stellt sich die Frage nach *Existenz* und *Eindeutigkeit* von $\mathrm{ggT}(x, y)$. Wir beweisen diese Eigenschaften im Anhang.

Wir bemerken, dass $\mathrm{ggT}(0, 0) = 0$ gilt. (Sei $d = \mathrm{ggT}(0, 0)$. Weil $0 \mid 0$, folgt mit Def. 4.3(b) $0 \mid d$ und damit $d = 0$.) Wenn $x \neq 0$ oder $y \neq 0$ gilt, kann es keinen gemeinsamen Teiler geben, der größer als $\max\{|x|, |y|\}$ ist, und der größte gemeinsame Teiler ist auch größtmöglich im Sinn der gewöhnlichen Ordnung auf $\mathbb{Z}$. Weil das Vorzeichen für die Teilbarkeit irrelevant ist, gilt stets $\mathrm{ggT}(x, y) = \mathrm{ggT}(|x|, |y|)$, und man kann sich immer auf den Fall nichtnegativer Argumente beschränken. Weiter gilt

$$\mathrm{ggT}(x, y) = \mathrm{ggT}(x + uy, y), \text{ für beliebige } x, y, u \in \mathbb{Z}. \tag{4.1}$$

(Wenn d gemeinsamer Teiler von x und y ist, dann teilt d auch $x + uy$. Wenn d gemeinsamer Teiler von $z = x + uy$ und y ist, dann teilt d auch $z - uy = x$. Also haben die Paare (x, y) und $(x + uy, y)$ dieselbe Menge gemeinsamer Teiler, und es folgt $\mathrm{ggT}(x, y) = \mathrm{ggT}(x + uy, y)$.)

Es gibt einen effizienten Algorithmus zur Ermittlung des größten gemeinsamen Teilers. Er beruht auf den Gleichungen

(i) $\mathrm{ggT}(x, y) = \mathrm{ggT}(|x|, |y|)$ für alle $x, y \in \mathbb{Z}$,

(ii) $\mathrm{ggT}(x, y) = \mathrm{ggT}(y, x)$ für alle $x, y \in \mathbb{Z}$,

(iii) $\mathrm{ggT}(a, 0) = a$ für $a \geq 0$,

(iv) $\mathrm{ggT}(a, b) = \mathrm{ggT}(b, a \bmod b)$ für $a \geq b > 0$.

((i) gilt, weil Teilbarkeit das Vorzeichen ignoriert. (ii) ist trivial. (iii) folgt daraus, dass jede Zahl Teiler von 0 ist. (iv) folgt aus (4.1) und (ii), weil $a \bmod b = a - qb$ mit $q = \lfloor a/b \rfloor$ gilt.)

Wir setzen die Beobachtung in ein iteratives Verfahren um.

Algorithmus 4.1 ***Euklidischer Algorithmus***
Input: Zwei ganze Zahlen x und y.
Methode:

```
1       a, b: integer; a ← |x|; b ← |y|;
2       while b > 0 repeat
3            (a, b) ← (b, a mod b);      // simultane Zuweisung
4       return a.
```

Die eigentliche Rechnung findet in der **while**-Schleife statt. In dieser Schleife wird immer ein Zahlenpaar durch ein anderes ersetzt, das dieselben gemeinsamen Teiler hat wie x und y. Wenn der Algorithmus terminiert, weil der Inhalt b von `b` Null geworden ist, kann man den Inhalt von `a` ausgeben.

Beispiel: Auf Eingabe $x = 10534$, $y = 12742$ ergibt sich der in Tab. 1 angegebene Ablauf. Die Zahlen a_i und b_i bezeichnen den Inhalt der Variablen `a` und `b`, nachdem die Schleife in Zeilen 2–3 i-mal ausgeführt worden ist. Die Ausgabe ist $46 = \mathrm{ggT}(10534, 12742)$.

Fakt 4.4
Algorithmus 4.1 gibt $\mathrm{ggT}(x, y)$ aus.

Wenn $|x| < |y|$, hat der erste Schleifendurchlauf nur den Effekt, die beiden Zahlen zu vertauschen. Wir ignorieren diesen trivialen Schleifendurchlauf. Wir betrachten die Zahlen a in `a` und b in `b`. Es gilt stets $a > b$, und b nimmt in jeder Runde strikt ab, also terminiert der Algorithmus. Um einzusehen, dass er sogar sehr schnell terminiert, bemerken wir Folgendes. Betrachte den Beginn eines Schleifendurchlaufs. Der Inhalt von `a` sei a, der Inhalt von `b` sei b, mit $a \geq b > 0$. Nach einem Schleifendurchlauf enthält `a` den Wert $a' = b$ und `b` den Wert $b' = a \bmod b$. Falls $b' = 0$, endet der Algorithmus. Sonst wird noch ein Durchlauf ausgeführt, an dessen Ende `a` den Wert $b' = a \bmod b$ enthält. Wir behaupten: $b' < \frac{1}{2}a$. Um dies zu beweisen, betrachten wir zwei Fälle: Wenn $b > \frac{1}{2}a$ ist, gilt $b' = a \bmod b = a - b < \frac{1}{2}a$. Wenn $b \leq \frac{1}{2}a$ ist, gilt $b' = a \bmod b < b \leq \frac{1}{2}a$. – Also wird der Wert in `a` in jeweils zwei Durchläufen mindestens halbiert. Nach dem ersten Schleifendurchlauf enthält `a` den Wert $\min\{x, y\}$. Daraus ergibt sich Teil (a) der folgenden Aussage.

i	a_i	b_i
0	10534	12742
1	12742	10534
2	10534	2208
3	2208	1702
4	1702	506
5	506	184
6	184	138
7	138	46
8	46	0

Tabelle 1: Ablauf des Euklidischen Algorithmus auf einem Beispiel

Fakt 4.5

(a) Die Schleife in Zeilen 2–3 wird höchstens $O(\log(\min\{x, y\}))$-mal ausgeführt.

(b) Die gesamte Anzahl von Ziffernoperationen für den Euklidischen Algorithmus ist $O((\log x)(\log y))$.

Man beachte, dass $\lceil \log(x+1) \rceil \approx \log x$ die Anzahl der Bits in der Binärdarstellung von x ist. Damit hat der Euklidische Algorithmus bis auf einen konstanten Faktor denselben Aufwand wie die Multiplikation von x und y, wenn man die Schulmethode benutzt. (Der Beweis der Schranke in (b) benötigt eine Rechnung, die die Längen der beteiligten Zahlen genauer verfolgt.)

Beispiel: (a) 21 und 25 sind teilerfremd. Es gilt $31 \cdot 21 + (-26) \cdot 25 = 651 - 650 = 1$.
(b) Auch -21 und 25 sind teilerfremd. Aus (a) folgt sofort $(-31)\cdot(-21)+(-26)\cdot 25 = 651 - 650 = 1$.
(c) Es gilt $\mathrm{ggT}(21, 35) = 7$, und $2 \cdot 35 - 3 \cdot 21 = 7$.

Die folgende sehr nützliche Aussage verallgemeinert diese Beobachtung:

Lemma 4.6 *... von Bezout*

(a) Wenn $x, y \in \mathbb{Z}$ teilerfremd sind, gibt es $s, t \in \mathbb{Z}$ mit $sx + ty = 1$.

(b) Für $x, y \in \mathbb{Z}$ gibt es $s, t \in \mathbb{Z}$ mit $sx + ty = \mathrm{ggT}(x, y)$.

Wir geben einen Algorithmus an, der zu x und y die Werte s und t (sehr effizient) berechnet. Damit ist die Frage der Existenz natürlich gleich mit erledigt. Vorab bemerken wir noch, dass es eine Art Umkehrung von (a) gibt: Wenn $sx + ty = 1$ für ganze Zahlen s und t gilt, dann sind x und y teilerfremd. (*Beweis*: Alle gemeinsamen Teiler von x und y teilen auch 1, sind also 1 oder -1. Daraus folgt $\mathrm{ggT}(x, y) = 1$.)

Für den Algorithmus können wir o. B. d. A. annehmen, dass $x, y \geq 0$ gilt. Die Umrechnung für negative Inputs ist offensichtlich.

Algorithmus 4.2 ***Erweiterter Euklidischer Algorithmus***
EINGABE: Natürliche Zahlen x und y.
METHODE:

```
1    a, b, sa, ta, sb, tb, q: integer;
2    a ← x; b ← y;
3    sa ← 1; ta ← 0; sb ← 0; tb ← 1;
4    while b > 0 repeat
5        q ← a div b;
6        (a, b) ← (b, a − q · b);
7        (sa, ta, sb, tb) ← (sb, tb, sa − q · sb, ta − q · tb);
8    return (a, sa, ta);
```

Genau wie im ursprünglichen Euklidischen Algorithmus findet die eigentliche Arbeit in der **while**-Schleife (Zeilen 4–7) statt.

Die Idee hinter dem Algorithmus ist folgende. Wie im (einfachen) Euklidischen Algorithmus werden in den Variablen `a` und `b` Zahlen a und b mitgeführt, die stets $\mathrm{ggT}(a, b) = d = \mathrm{ggT}(x, y)$ erfüllen. Nach dem ersten Durchlauf gilt $b \leq a$. Die Variablen `sa`, `ta`, `sb` und `tb` enthalten immer Zahlenpaare (s_a, t_a) und (s_b, t_b), die folgende Gleichungen erfüllen:

$$\begin{aligned} a &= s_a \cdot x + t_a \cdot y, \\ b &= s_b \cdot x + t_b \cdot y. \end{aligned} \tag{4.2}$$

Diese Gleichung wird durch die Initialisierung hergestellt. In einem Schleifendurchlauf wird a durch b ersetzt und (s_a, t_a) durch (s_b, t_b), und es wird b durch $a - q \cdot b$ ersetzt sowie (s_b, t_b) durch $(s_a - q \cdot s_b, t_a - q \cdot t_b)$. Dadurch bleiben die Gleichungen (4.2) gültig. Wenn schließlich $b = 0$ geworden ist, gilt $d = \mathrm{ggT}(x, y) = a = s_a \cdot x + t_a \cdot y$. Das bedeutet, dass die Ausgabe das gewünschte Ergebnis darstellt.

Als Beispiel betrachten wir den Ablauf des Algorithmus auf der Eingabe $(x, y) = (10534, 12742)$. Die Zahlen a_i ,b_i, $s_{a,i}$, $t_{a,i}$, $s_{b,i}$, $t_{b,i}$ bezeichnen den Inhalt von `a`, `b`, `sa`, `ta`, `sb`, `tb` nach dem i-ten Schleifendurchlauf.

i	a_i	b_i	$s_{a,i}$	$t_{a,i}$	$s_{b,i}$	$t_{b,i}$	q_i
0	10534	12742	1	0	0	1	–
1	12742	10534	0	1	1	0	–
2	10534	2208	1	0	−1	1	1
3	2208	1702	−1	1	5	−4	4
4	1702	506	5	−4	−6	5	1
5	506	184	−6	5	23	−19	3
6	184	138	23	−19	−52	43	2
7	138	46	−52	43	75	−62	1
8	46	0	75	−62	−277	229	3

Die Ausgabe ist $(46, 75, -62)$. Man überprüft leicht, dass

$$46 = \mathrm{ggT}(10534, 12742) = 75 \cdot 10534 - 62 \cdot 12742$$

gilt. – Allgemein gilt:

Fakt 4.7
Wenn Algorithmus 4.2 auf Eingabe (x, y) mit $x, y \geq 0$ gestartet wird, dann gilt:

(a) Für die Ausgabe (d, s, t) gilt $d = \mathrm{ggT}(x, y) = sx + ty$.

(b) Die Anzahl der Schleifendurchläufe ist dieselbe wie beim gewöhnlichen Euklidischen Algorithmus.

(c) Die Anzahl von Ziffernoperationen für Algorithmus 4.2 ist $O((\log x)(\log y))$.

Wir notieren noch eine wichtige Folgerung aus dem Lemma von Bezout.[3]

[3]Die Aussage ist aus der Schule bekannt: Wenn eine Zahl z. B. durch 3 und durch 5 teilbar ist, dann ist sie auch durch 15 teilbar. Dort benutzt man die Primzahlzerlegung zur Begründung. Diese ist aber gar nicht nötig.

Fakt 4.8
Wenn x und y teilerfremd sind und a sowohl durch x als auch durch y teilbar ist, dann ist a auch durch xy teilbar.

Beweis: Weil x und y Teiler von a sind, kann man $a = ux$ und $a = vy$ schreiben, für ganze Zahlen u, v. Weil x und y teilerfremd sind, liefert Lemma 4.6(a) zwei ganze Zahlen s und t mit $1 = sx + ty$. Dann ist

$$a = asx + aty = vysx + uxty = (vs + ut)xy,$$

also ist xy Teiler von a. □

4.2 Modulare Arithmetik

Definition 4.9
Für $m \geq 2$ definieren wir eine zweistellige Relation auf $\mathbb{Z}$:

$$x \equiv y \pmod{m} \quad \text{heißt} \quad m \mid (x - y).$$

Man sagt: „x ist kongruent zu y modulo m.“ In der Mathematik sieht man auch oft die kompaktere Notation $x \equiv y\ (m)$ oder $x \equiv_m y$. Es besteht eine enge Beziehung zwischen dieser Relation und der Division mit Rest.

Fakt 4.10 (i) $x \equiv y \pmod{m}$ gilt genau dann wenn $x \bmod m = y \bmod m$ gilt.

(ii) Die zweistellige Relation $\cdot \equiv \cdot \pmod{m}$ ist eine Äquivalenzrelation, sie ist also reflexiv, transitiv und symmetrisch.

Beispiel für (i): $29 \bmod 12 = 53 \bmod 12 = 5$ und $53 - 29 = 24$ ist durch 12 teilbar.

Der *Beweis* von (i) ist eine leichte Übung; (ii) folgt sofort aus (i).

Die Kongruenzrelation $\cdot \equiv \cdot \pmod{m}$ führt (wie jede Äquivalenzrelation) zu einer Zerlegung der Grundmenge $\mathbb{Z}$ in Äquivalenzklassen (die hier „Restklassen“ heißen):

$$[x]_m = [x] = \{y \in \mathbb{Z} \mid x \equiv y \pmod{m}\} = \{y \in \mathbb{Z} \mid x \bmod m = y \bmod m\}.$$

Wir definieren: $m\mathbb{Z} := \{\ldots, -3m, -2m, -m, 0, m, 2m, 3m, \ldots\}$ und
$x + A := \{x + y \mid y \in A\}$, für $A \subseteq \mathbb{Z}$.

Beispiel: Für $m = 3$ gibt es die drei Restklassen

$$\begin{aligned}[0] = [0]_3 &= \{\ldots, -6, -3, 0, 3, 6, \ldots\} = 0 + 3\mathbb{Z},\\ [1] = [1]_3 &= \{\ldots, -5, -2, 1, 4, 7, \ldots\} = 1 + 3\mathbb{Z},\\ [2] = [2]_3 &= \{\ldots, -4, -1, 2, 5, 8, \ldots\} = 2 + 3\mathbb{Z}.\end{aligned}$$

Mit den Restklassen kann man dann wieder rechnen: Addition und Multiplikation lassen sich wie folgt definieren.

$$\begin{aligned}[x]_m + [y]_m &:= [x + y]_m,\\ [x]_m \cdot [y]_m &:= [x \cdot y]_m.\end{aligned}$$

Beispielsweise gelten für $m = 3$ die Gleichheiten $[4] + [5] = [9] = [0]$ und $[4] \cdot [5] = [20] = [2]$.

Fakt 4.11
Diese Operationen sind *wohldefiniert*, d. h., aus $x \equiv x' \pmod m$ und $y \equiv y' \pmod m$ folgt $[x + y]_m = [x' + y']_m$ und $[x \cdot y]_m = [x' \cdot y']_m$.

(Der Beweis ist einfach. Weil $x \equiv x' \pmod m$ und $y \equiv y' \pmod m$ gilt, sind $x - x'$ und $y - y'$ durch m teilbar. Also ist auch $xy - x'y' = x(y - y') + (x - x')y'$ durch m teilbar, und es gilt $x \cdot y \equiv x' \cdot y' \pmod m$. Der Fall der Addition ist noch einfacher.)

Aus der Definition und der Wohldefiniertheit ergibt sich, dass man anstatt mit Restklassen auch mit Repräsentanten rechnen kann. Statt $([5]_3 \cdot [5]_3) \cdot [2]_3 = [25]_3 \cdot [2]_3 = [1]_3 \cdot [2]_3 = [2]_3$ schreibt man dann einfach

$$(5 \cdot 5) \cdot 2 \equiv 25 \cdot 2 \equiv 1 \cdot 2 \equiv 2 \pmod 3.$$

Fakt 4.11 besagt auch, dass an jeder Stelle einer solchen Rechnung jede Zahl durch eine dazu kongruente Zahl ersetzt werden darf, je nachdem, wie es bequem ist. Beispiel: $(5 \cdot 5) \cdot 2 \equiv ((-1) \cdot (-1)) \cdot (-1) = (-1)^3 = -1 \equiv 2 \pmod 3$. Da $(x \bmod m) \equiv x \pmod m$ für alle x und $m \geq 1$ gilt, kann man in „modulo-m-Rechnungen“ eine Zahl x insbesondere immer durch ihren Rest modulo m ersetzen.

Beispiel: Um $13^7 \bmod 11$ zu berechnen, rechnet man $13^7 \equiv 2^7 \equiv 2^5 \cdot 4 = 32 \cdot 4 \equiv (-1) \cdot 4 = -4 \equiv 7 \pmod{11}$. Um $3^{1006} \bmod 7$ zu berechnen, bemerkt man, dass $3^2 \bmod 7 = 2$ ist, also $3^{1006} \equiv 2^{503} \pmod 7$. Weil nun $2^3 \bmod 7 = 1$ gilt, folgt $2^{503} = (2^3)^{167} \cdot 2^2 \equiv 1^{167} \cdot 4 = 4 \pmod 7$.

Zu $m \geq 1$ betrachtet die Menge aller Restklassen:[4]

$$\mathbb{Z}_m := \mathbb{Z}/m\mathbb{Z} := \{[x]_m \mid x \in \mathbb{Z}\} = \{[x] \mid 0 \leq x < m\}.$$

Solange es nicht zu Missverständnissen führt, schreibt man auch $\mathbb{Z}_m = \{x \mid 0 \leq x < m\}$, zusammen mit „Addition modulo m“ und „Multiplikation modulo m“. Damit meint man, dass man mit den Repräsentanten der Restklassen rechnet, die in $\{0, 1, \ldots, m-1\}$ liegen.

Fakt 4.12
Für jedes $m \geq 2$ bildet die Menge $\mathbb{Z}_m$ mit den Operationen *Addition modulo m* und *Multiplikation modulo m* einen *kommutativen Ring* mit 1.

Das heißt im Detail: Die Operationen $+ \pmod m$ und $\cdot \pmod m$ führen nicht aus dem Bereich $\mathbb{Z}_m$ heraus. Die Addition erfüllt alle Rechenregeln für kommutative Gruppen, d. h. sie ist kommutativ und assoziativ, es gibt ein neutrales Element, nämlich $[0]$, und zu jedem $[x]$ gibt es ein Inverses $-[x] = [-x]$. (Es gilt ja $[x] + [-x] = [0]$. Beachte: Für $0 < x < m$ gilt $0 < m - x < m$ und $[x] + [m - x] = [m] = [0]$, also $-[x] = [m-x]$. Insbesondere ist $-[1] = [-1] = [m-1]$ das additive Inverse zu $[1]$.) Die Multiplikation ist assoziativ und kommutativ, und sie hat $[1]$ als neutrales Element ($[1]\cdot[x] = [x]\cdot[1] = [x]$); für Addition und Multiplikation gelten die Distributivgesetze. (Siehe auch Bemerkung A.1 im Anhang.)

Lemma 4.13
Für jedes $m \geq 2$ ist die Abbildung $\mathbb{Z} \to \mathbb{Z}_m$, $x \mapsto [x]$, ein Homomorphismus; d. h. für $x, y \in \mathbb{Z}$ gilt: $[x + y] = [x] + [y]$ und $[x \cdot y] = [x] \cdot [y]$.

(Die folgt direkt aus Definition der Operationen.)

In vielen Anwendungen der Zahlentheorie in kryptographischen Verfahren kommt es darauf an, Potenzen $x^y \bmod m$ zu berechnen. Dabei kann $y \geq 0$ eine sehr große Zahl sein. Beispiel:

$$3^{1384788374932954500363985493554603584759389} \bmod 28374618732464817362847326847331872341234$$

Wieso kann ein Computer das Ergebnis (18019019948605604024937414441328931495971) in Bruchteilen von Sekunden berechnen? Auf keinen Fall kann er y Multiplikationen

[4] $\mathbb{Z}/m\mathbb{Z}$ liest man mathematisch korrekt als „Zet modulo em-Zet“. $\mathbb{Z}_m$ ist einfach „Zet-em“.

durchführen. Die folgende einfache rekursive Formel weist den Weg zu einer effizienten Berechnung:

$$x^y \bmod m = \begin{cases} 1 & \text{, wenn } y = 0, \\ x \bmod m & \text{, wenn } y = 1, \\ ((x^2 \bmod m)^{y/2}) \bmod m & \text{, wenn } y \geq 2 \text{ gerade ist}, \\ (((x^2 \bmod m)^{(y-1)/2} \bmod m) \cdot x) \bmod m & \text{, wenn } y \geq 2 \text{ ungerade ist.} \end{cases}$$

Man beachte noch, dass

$$\lfloor y/2 \rfloor = \begin{cases} y/2 & \text{für gerade } y, \\ (y-1)/2 & \text{für ungerade } y. \end{cases}$$

Diese Formeln führen unmittelbar zu folgender rekursiver Prozedur.

Algorithmus 4.3 ***Schnelle modulare Exponentiation, rekursiv***
function modexp(x, y, m)
EINGABE: Ganze Zahlen x, $y \geq 0$, and $m \geq 1$, mit $0 \leq x < m$.
METHODE:
0 **if** $y = 0$ **then return** 1;
1 **if** $y = 1$ **then return** x;
2 $\mathtt{z} \leftarrow \text{modexp}((x \cdot x) \bmod m, \lfloor y/2 \rfloor, m)$; // rekursiver Aufruf
3 **if** y ist ungerade **then** $\mathtt{z} \leftarrow (\mathtt{z} \cdot x) \bmod m$
4 **return** z.

Man erkennt sofort, dass in jeder Rekursionsebene die Bitanzahl des Exponenten y um 1 sinkt, dass also die Anzahl der Rekursionsebenen etwa $\log y$ beträgt. In jeder Rekursionsstufe ist eine oder sind zwei Multiplikationen modulo m auszuführen, was $O((\log m)^2)$ Ziffernoperationen erfordert (Schulmethode).

Beispiel: Wir berechnen $13^{43} \bmod 19$.

i		$x^{2^i} \bmod 19$	$\lfloor y/2^i \rfloor$	Faktor (wenn $\lfloor y/2^i \rfloor$ ungerade)
0	x	13	43	13
1	x^2	$13^2 \equiv (-6)^2 \equiv 36 \equiv 17$	21	17
2	x^4	$17^2 \equiv (-2)^2 = 4$	(gerade) 10	–
3	x^8	$4^2 = 16$	5	16
4	x^{16}	$16^2 \equiv (-3)^2 = 9$	(gerade) 2	–
5	x^{32}	$9^2 \equiv (-10)^2 = 100 \equiv 5$	1	5

Produkt: $x \cdot x^2 \cdot x^8 \cdot x^{16} \cdot x^{32} \equiv 13 \cdot 17 \cdot 16 \cdot 5 \equiv (-6)(-2)(-3)5 = -180 \equiv -9 \equiv 10$.

Lemma 4.14
Sei $x < m$. Die Berechnung von $x^y \bmod m$ benötigt $O(\log y)$ Multiplikationen und Divisionen modulo m von Zahlen aus $\{0, \ldots, m^2 - 1\}$, und damit $O((\log y)(\log m)^2)$ Ziffernoperationen.

Bemerkung: Man kann denselben Algorithmus in einem beliebigen *Monoid* $(M, \circ, e)$ ($M \neq \emptyset$ ist eine Menge, $\circ\colon M \times M \to M$ ist assoziative Operation mit neutralem Element $e \in M$) benutzen. Monoide bilden zum Beispiel:

- $(\mathbb{Z}_m, \cdot_m, 1)$, wo $\cdot_m$ die Multiplikation modulo m ist;
- $(\mathbb{N}, +, 0)$: die natürlichen Zahlen mit der Addition (neutral: 0);
- $(\mathbb{N}, \cdot, 1)$: die natürlichen Zahlen mit der Multiplikation (neutral: 1);
- quadratische Matrizen über einem Ring mit 1, mit Matrixmultiplikation (neutral: Einheitsmatrix);
- die Menge Σ^* aller Wörter über einem Alphabet Σ, mit der Konkatenation (neutral: das leere Wort);
- jede Gruppe $(G, \circ, e)$ (G ist die Grundmenge, $\circ$ die Operation und e das neutrale Element.)

Wenn man nur eine assoziative Operation und kein neutrales Element hat, funktioniert der Algorithmus für Exponenten $y \geq 1$.

4.3 Inverse in Restklassenringen

Wir untersuchen nun, wie es mit multiplikativen Inversen im Ring $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$ steht. Das heißt: Gegeben $x \in \mathbb{Z}_m$, wann gibt es ein $y \in \mathbb{Z}_m$ mit $xy \mod m = 1$? Wenn $x = 0$, geht das natürlich nie. Für $1 \leq x < m$ muss man genauer hinsehen.

Zunächst eine einfache Beobachtung:

Lemma 4.15
Für jedes $m \geq 2$ und alle $x, y \in \mathbb{Z}$ gilt*:* Wenn $x \equiv y \pmod m$, dann gilt $\mathrm{ggT}(x, m) = \mathrm{ggT}(y, m)$. (In Worten: Der größte gemeinsame Teiler von x und m hängt nur von der Restklasse $[x]$ modulo m ab.) Insbesondere gilt $\mathrm{ggT}(x, m) = \mathrm{ggT}(x \bmod m, m)$.

Beweis: Sei $x = y + am$. Dann ist jeder gemeinsame Teiler von x und m auch gemeinsamer Teiler von y und m und umgekehrt. □

Wegen dieses Lemmas kann man auch unbesorgt $\mathrm{ggT}(x, m)$ für Elemente x von $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$ schreiben. In diesem Ring spielen die Elemente, die ein *multiplikatives* Inverses haben, eine besondere Rolle.

Beispiel: Bei $m = 15$ gilt $1 \cdot 1 = 1$, $2 \cdot 8 = 16 \equiv 1$, $4 \cdot 4 = 16 \equiv 1$, $7 \cdot 13 = 91 \equiv 1$, $11 \cdot 11 = 121 \equiv 1$, $14 \cdot 14 \equiv (-1)^2 = 1 \pmod{15}$. Bei den Zahlen $0, 3, 5, 6, 9, 10, 12$ findet man kein multiplikatives Inverses. (Beispiel: Jede Zahl $12 \cdot y - q \cdot 15$ ist durch 3 teilbar, also kann $12 \cdot y \bmod 15$ für kein y gleich 1 sein.) Daher haben in $\mathbb{Z}_{15}$ die acht Zahlen $1, 2, 4, 7, 8, 11, 13, 14$ ein multiplikatives Inverses modulo 15, die anderen sieben Zahlen haben keines. – Die Elemente von $\mathbb{Z}_{15}$ mit einem multiplikativen Inversen sind genau die, die zu 15 teilerfremd sind.

Fakt 4.16
Für jedes $m \geq 2$ und jedes $x \in \mathbb{Z}$ gilt*:*
Es gibt ein y mit $xy \bmod m = 1$ genau dann wenn $\mathrm{ggT}(x, m) = 1$.

Beweis:
„⇒“: Es sei $xy \bmod m = 1$. Das heißt: Es gibt ein $q \in \mathbb{Z}$ mit $xy - qm = 1$. Dann teilt jeder gemeinsame Faktor von x und m auch 1, also sind x und m teilerfremd.

„⇐“: x und m seien teilerfremd. Nach dem Lemma von Bezout (Lemma 4.6(a)) gibt es $s, t \in \mathbb{Z}$ mit $sx + tm = 1$. Setze $y := s \bmod m$. Dann gilt:

$$(x \cdot y) \bmod m = (x \cdot (s \bmod m)) \bmod m = sx \bmod m = 1. \qquad \square$$

Beispiel: $x = 22$ und $m = 15$ sind teilerfremd. Mit dem erweiterten Euklidischen Algorithmus findet man $s = -2$ und $t = 3$, so dass $sx + tm = (-2) \cdot 22 + 3 \cdot 15 = -44 + 45 = 1$ gilt. Setze $y = (-2) \bmod 15 = 13$. Man kontrolliert: $22 \cdot 13 = 286 = 19 \cdot 15 + 1 \equiv 1 \bmod 15$.

Wir bemerken allgemein: Mit dem erweiterten Euklidischen Algorithmus 4.2 berechnet man leicht d und Koeffizienten $s, t \in \mathbb{Z}$ mit $sx + tm = d = \mathrm{ggT}(x, m)$. Wenn $d > 1$ ist, gibt es kein Inverses zu x in $\mathbb{Z}_m$. Wenn $d = 1$ ist, folgt $sx \bmod m = 1$, also ist $s \bmod m$ das gewünschte inverse Element. Die Rechenzeit für das Berechnen des „modularen Inversen“ beträgt also $O((\log x)(\log m))$.

Die Menge der invertierbaren Elemente von $\mathbb{Z}_m$ erhält eine eigene Bezeichnung.

Definition 4.17
Für $m \geq 2$ sei $\mathbb{Z}_m^* := \{x \in \mathbb{Z}_m \mid \mathrm{ggT}(x, m) = 1\}$.

(Wieder sind eigentlich die Restklassen $[x]_m$, $0 \leq x < m$, $\mathrm{ggT}(x, m) = 1$, gemeint.)

Fakt 4.18 (∗)
Für jedes $m \geq 2$ gilt*:*
$\mathbb{Z}_m^*$ mit der Multiplikation modulo m als Operation ist eine (kommutative) Gruppe.

Beispiel: $\mathbb{Z}_{21}^* = \{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$ und $\mathbb{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$.

Wir haben $8 \cdot 16 \equiv 128 \equiv 2 \pmod{21}$, mit $2 \in \mathbb{Z}_{21}^*$. Für $17 \in \mathbb{Z}_{21}^*$ gibt es das Inverse 5, denn $17 \cdot 5 = 85 \equiv 1 \pmod{21}$.

Aus den Gruppeneigenschaften weiß man, dass Inverse eindeutig bestimmt sind. Für das Inverse von x (wenn es existiert) schreiben wir $x^{-1} \bmod m$ (gemeint ist die Restklasse oder der eindeutig bestimmte Repräsentant in $\{0, 1, \ldots, m-1\}$).

Am schönsten ist die Situation, wenn alle Zahlen $1, \ldots, m-1$ in $\mathbb{Z}_m^*$ liegen. Das heißt insbesondere, dass keine der Zahlen $2, 3, \ldots, m-1$ die Zahl m teilt. Um diese Situation zu beschreiben, definieren wir vorläufig den Begriff der Primzahl. (Eine genauere Diskussion von Primzahlen folgt in Abschnitt 4.5.) Man erinnere sich, dass jede ganze Zahl x durch x und $-x$ sowie durch 1 und -1 teilbar ist.

> Eine ganze Zahl $p \geq 1$ heißt **Primzahl**, wenn p genau zwei positive Teiler hat, nämlich 1 und p.

Die Folge der Primzahlen beginnt mit $2, 3, 5, 7, 11, 13, 17, 19, 23, \ldots$.

Fakt 4.19 (∗)
Für jedes $m \geq 2$ sind folgende Aussagen äquivalent*:*

(a) m ist eine Primzahl.

(b) $\mathbb{Z}_m^* = \{1, \ldots, m-1\}$.

(c) $\mathbb{Z}_m$ ist ein Körper.

Der *Beweis* erfolgt durch einen Ringschluss. „(a) ⇒ (b)“: Sei m Primzahl. Dann kann für kein Element $x \in \{1, \ldots, m-1\}$ die Beziehung $\mathrm{ggT}(x, m) > 1$ gelten, weil sonst die Zahl $\mathrm{ggT}(x, m)$ ein Teiler von m strikt zwischen 1 und m wäre. „(b) ⇒ (c)“: Wenn $\mathbb{Z}_m^* = \{1, \ldots, m-1\}$ gilt, hat nach Fakt 4.16 jedes Element von $\mathbb{Z}_m - \{0\}$ ein multiplikatives Inverses. Das ist genau die Eigenschaft, die dem Ring $\mathbb{Z}_m$ zum Körper fehlt. „(c) ⇒ (a)“: Das beweisen wir durch Kontraposition. Sei also (a) falsch, d. h. sei m keine Primzahl. Dann gibt es ein $x \in \{2, \ldots, m-1\}$, das Teiler von m ist. Insbesondere ist $\mathrm{ggT}(x, m) = x > 1$. Mit Fakt 4.16 folgt, dass x kein multiplikatives Inverses modulo m hat, also ist $\mathbb{Z}_m$ kein Körper, d. h. (c) ist falsch. □

Beispiel: $m = 13$. Wir geben für jedes $x \in \mathbb{Z}_{13}^*$ das Inverse y sowie das Produkt $x \cdot y$ an (das natürlich bei der Division durch 13 Rest 1 lassen muss).

x	1	2	3	4	5	6	7	8	9	10	11	12
y	1	7	9	10	8	11	2	5	3	4	6	12
$x \cdot y$	1	14	27	40	40	66	14	40	27	40	66	144

14 ist keine Primzahl, und es gibt keine Zahl y mit $2 \cdot y \bmod 14 = 1$; das heißt, dass $2 \notin \mathbb{Z}_{14}^*$ und daher, dass $\mathbb{Z}_{14}$ kein Körper ist.

Wir notieren noch einen altehrwürdigen[5] Satz aus der Zahlentheorie.

Fakt 4.20 ***(Kleiner Satz von Fermat)***
Wenn p eine Primzahl ist, dann gilt*:*

$$a^{p-1} \bmod p = 1, \text{ für jedes } a \in \mathbb{Z}_p^*.$$

Beweis: Sei $a \in \mathbb{Z}_p^*$ gegeben. Betrachte die Abbildung

$$g_a \colon \mathbb{Z}_p^* \ni s \mapsto as \bmod p \in \mathbb{Z}_p^*.$$

Diese Abbildung ist injektiv, da für das zu a inverse Element $b = a^{-1} \bmod p$ gilt: $b \cdot g_a(s) \bmod p = b(as) \bmod p = ((ba) \bmod p)s \bmod p = s$. Also gilt:

$$\{1, \ldots, p-1\} = \{g_a(1), \ldots, g_a(p-1)\}.$$

Wir multiplizieren die Zahlen $1, \ldots, p-1$ in zwei Anordnungen:

$$\underbrace{1 \cdot \ldots \cdot (p-1) \bmod p}_{=X} = g_a(1) \cdot \ldots \cdot g_a(p-1) \bmod p = a^{p-1} \cdot \underbrace{(1 \cdot \ldots \cdot (p-1) \bmod p)}_{=X}.$$

Wenn wir beide Seiten mit dem multiplikativen Inversen von $X := 1 \cdot \ldots \cdot (p-1) \bmod p$ multiplizieren, erhalten wir $1 = a^{p-1} \bmod p$. □

Wir bemerken, dass auch eine gewisse Umkehrung gilt, sogar für beliebige m: Wenn $a^{m-1} \bmod m = 1$ ist, d. h. $a^{m-1} - qm = 1$ für ein q, dann folgt $\mathrm{ggT}(a, m) = 1$. Wenn also $a \in \mathbb{Z}_m - \mathbb{Z}_m^*$, dann gilt auf jeden Fall $a^{m-1} \bmod m \neq 1$.

[5]Der Satz wurde von Pierre de Fermat, 1607–1665, einem französischen Mathematiker und Juristen, gefunden.

4.4 Der Chinesische Restsatz

Der „Chinesische Restsatz“ besagt im Wesentlichen, dass für teilerfremde Zahlen m und n die Strukturen $\mathbb{Z}_m \times \mathbb{Z}_n$ (mit komponentenweisen Operationen) und $\mathbb{Z}_{mn}$ isomorph sind.

Wir beginnen mit einem Beispiel, nämlich $m = 3$, $n = 8$, also $mn = 24$. Die folgende Tabelle gibt die Reste der Zahlen $x \in \{0, 1, \ldots, 23\}$ modulo 3 und modulo 8 an. Die Restepaare wiederholen sich zyklisch für andere $x \in \mathbb{Z}$.

x	0	1	2	3	4	5	6	7	8	9	10	11
x mod 3	0	1	2	0	1	2	0	1	2	0	1	2
x mod 8	0	1	2	3	4	5	6	7	0	1	2	3

x	12	13	14	15	16	17	18	19	20	21	22	23
x mod 3	0	1	2	0	1	2	0	1	2	0	1	2
x mod 8	4	5	6	7	0	1	2	3	4	5	6	7

Wenn wir die Einträge in Zeilen 2 und 3 als 24 Paare in $\mathbb{Z}_3 \times \mathbb{Z}_8$ ansehen, erkennen wir, dass sie alle verschieden sind, also auch alle Möglichkeiten in $\{0, 1, 2\} \times \{0, 1, \ldots, 7\}$ abdecken. D. h.: Die Abbildung $x \mapsto (x \bmod 3, x \bmod 8)$ ist eine Bijektion zwischen $\mathbb{Z}_{24}$ und $\mathbb{Z}_3 \times \mathbb{Z}_8$. Zudem spiegeln sich arithmetische Operationen auf den Elementen von $\mathbb{Z}_{24}$ in den Resten modulo 3 und 8 wider. Beispielsweise liefert die Addition von $(2, 7)$ und $(2, 1)$ das Resultat $(1, 0)$, das der Addition von 23 und 17 mit dem Resultat $40 \bmod 24 = 16$ entspricht. Genauso ist

$$(2^5 \bmod 3, 3^5 \bmod 8) = (2, 3),$$

was der Beobachtung $11^5 \bmod 24 = 11$ entspricht.

Der Chinesische Restsatz[6] sagt im wesentlichen, dass eine solche strukturelle Entsprechung zwischen den Resten modulo mn und Paaren von Resten modulo m bzw. n immer gilt, wenn m und n teilerfremd sind.

[6] Die Bezeichnung hat sich eingebürgert. Eigentlich geht es in dem Satz um Rest*klassen*.

Fakt 4.21 ***Chinesischer Restsatz*** **(∗)**
m und n seien teilerfremd. Dann ist die Abbildung[a]

$$\Phi\colon \mathbb{Z}_{mn} \ni x \mapsto (x \bmod m, x \bmod n) \in \mathbb{Z}_m \times \mathbb{Z}_n$$

bijektiv. Weiterhin*:* Wenn $\Phi(x) = (x_1, x_2)$ und $\Phi(y) = (y_1, y_2)$, dann gilt*:*

(a) $\Phi(x +_{mn} y) = (x_1 +_m y_1, x_2 +_n y_2)$*;*

(b) $\Phi(x \cdot_{mn} y) = (x_1 \cdot_m y_1, x_2 \cdot_n y_2)$*;*

(c) $\Phi(1) = (1, 1)$.

(Dabei bezeichnen $+_j$ und $\cdot_j$ die Addition und die Multiplikation modulo j.)

[a] Φ ist der griechische Buchstabe *Phi*.

Für mathematisch-strukturell orientierte Leser/innen: Die Gleichungen (a) bis (c) kann man etwas abstrakter auch so fassen, dass die Abbildung Φ ein Ring-mit-1-Isomorphismus zwischen $\mathbb{Z}_{mn}$ und $\mathbb{Z}_m \times \mathbb{Z}_n$ ist.

Man kann sich noch fragen, wie man nötigenfalls zu gegebenen Zahlen $s \in \mathbb{Z}_m$ und $t \in \mathbb{Z}_n$ die Zahl $x \in \mathbb{Z}_{mn}$ *berechnen* kann, die $\Phi(x) = (s, t)$ erfüllt. Dazu betrachtet man zunächst den Fall $s = 1$ und $t = 0$. Weil m und n teilerfremd sind, kann man mit dem erweiterten Euklidischen Algorithmus ein $u \in \mathbb{Z}_m$ mit $un \bmod m = 1$ finden. Wir setzen $y = un \in \mathbb{Z}_{mn}$. Dann gilt $y \bmod m = 1$ und $y \bmod n = 0$. Analog findet man ein $z \in \mathbb{Z}_{mn}$ mit $z \bmod m = 0$ und $z \bmod n = 1$. Nun setzen wir $x := (sy{+}tz) \bmod mn \in \mathbb{Z}_{mn}$. Wir haben, modulo m gerechnet: $x \equiv sy{+}tz \equiv s{\cdot}1{+}t{\cdot}0 \equiv s \pmod m$. Analog ergibt sich $x \equiv sy{+}tz \equiv s{\cdot}0{+}t{\cdot}1 \equiv t \pmod n$, wie gewünscht. Der Berechnungsaufwand für das Finden von x ist $O((\log m)(\log n))$ Ziffernoperationen, das geht also sehr schnell.

Beispiel: $m = 5$, $n = 8$, $s = 3$, $t = 7$. Wir finden $u = 2$ mit $u \cdot 8 \bmod 5 = 1$ und $y = 2 \cdot 8 = 16$ sowie $v = 5$ mit $v \cdot 5 \bmod 8 = 1$ und $z = 5 \cdot 5 = 25$. Nun setzen wir $x = (3 \cdot 16 + 7 \cdot 25) \bmod 40 = (48 + 175) \bmod 40 = (8 + 15) \bmod 40 = 23$. Und tatsächlich: $23 \bmod 5 = 3$ und $23 \bmod 8 = 7$.

Wir wollen noch untersuchen, wie sich Zahlen, die zu m und n teilerfremd sind, in der Sichtweise des Chinesischen Restsatzes verhalten.

Proposition 4.22 **(∗)**
Wenn man die Abbildung Φ aus dem Chinesischen Restsatz auf $\mathbb{Z}^*_{mn}$ einschränkt, ergibt sich eine Bijektion zwischen $\mathbb{Z}^*_{mn}$ und $\mathbb{Z}^*_m \times \mathbb{Z}^*_n$.

Bemerkung: Der Chinesische Restsatz und die nachfolgenden Bemerkungen und Behauptungen lassen sich leicht auf $r > 2$ paarweise teilerfremde Faktoren $n_1, \ldots, n_r$ verallgemeinern. Die Aussagen lassen sich durch vollständige Induktion über r beweisen.

Mit Prop. 4.22 können wir eine übersichtliche Formel für die Kardinalitäten der Mengen $\mathbb{Z}_m^*$, $m \geq 2$, entwickeln.[7]

Definition 4.23 ***(Eulersche φ-Funktion)***
Für $m \geq 2$ sei

$$\varphi(m) := |\mathbb{Z}_m^*| = |\{x \mid 0 < x < m, \mathrm{ggT}(x, m) = 1\}|.$$

Einige Beispielwerte, die man durch Aufzählen findet, sind in Tab. 2 angegeben.

m	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\varphi(m)$	1	2	2	4	2	6	4	6	4	10	4	12	6	8	8	16	6	18	8

Tabelle 2: Eulersche φ-Funktion für kleine m

Folgendes ist eine unmittelbare Konsequenz aus Proposition 4.22:

Lemma 4.24
Für teilerfremde Zahlen n und m gilt $\varphi(mn) = \varphi(m) \cdot \varphi(n)$.

(Man teste $20 = 4 \cdot 5$ und $12 = 3 \cdot 4$.)

Wir können den kleinen Satz von Fermat (Fakt 4.20) auf den Fall beliebiger m verallgemeinern. Auch der Beweis ist sehr ähnlich.

Fakt 4.25 ***(Satz von Euler)*** **$(*)$**
Für $m \geq 2$ und x mit $\mathrm{ggT}(m, x) = 1$ gilt*:* $x^{\varphi(m)} \bmod m = 1$.

Wenn m viele verschiedene kleine Primfaktoren hat, kann $\varphi(m)$ auch deutlich kleiner sein als m, z. B. $\varphi(210) = \varphi(2 \cdot 3 \cdot 5 \cdot 7) = 1 \cdot 2 \cdot 4 \cdot 6 = 48$. Es gilt aber: $\varphi(m) = m - 1$, wenn m eine Primzahl ist, und $\varphi(m) \geq \pi(m) = |\{p \leq m \mid p \text{ ist Primzahl}\}|$, wenn m zusammengesetzt ist.

[7] φ ist der griechische Buchstabe *phi*.

4.5 Primzahlen

Jede positive ganze Zahl x ist durch 1 und durch x teilbar.

Definition 4.26 (a) Eine Zahl $p \geq 1$ heißt **Primzahl**, wenn p genau zwei positive Teiler hat.[a] Diese Teiler sind dann 1 und p.

(b) Eine Zahl $x \geq 1$ heißt **zusammengesetzt**, wenn sie einen Teiler y mit $1 < y < x$ besitzt.[b]

[a] Die Zahl 1 hat nur einen positiven Teiler, nämlich 1. Also ist 1 keine Primzahl.
[b] Die Zahl 1 besitzt keinen Teiler y mit $1 < y < 1$. Also ist 1 nicht zusammengesetzt.

Bemerkung: Ein Blick auf Abbildung 1 zeigt, welche besondere Rolle 1, -1, 0 und die Primzahlen spielen. Die Zahlen 1 und -1 bilden das Minimum in der Teilbarkeitsrelation, die Primzahlen sitzen unmittelbar darüber, und die zusammengesetzten Elemente liegen strikt über den Primzahlen. Die 0, als Maximum in der Ordnung, sitzt strikt über allen zusammengesetzten Zahlen.

Fakt 4.27
Wenn p eine Primzahl ist und $p \mid xy$ gilt, dann gilt $p \mid x$ oder $p \mid y$.

Beweis: Wenn $p \mid x$, sind wir fertig. Also können wir $p \nmid x$ annehmen. Das heißt, dass $\mathrm{ggT}(p, x) = 1$ ist. Nach dem Lemma von Bezout können wir $1 = sp + tx$ schreiben, für ganze Zahlen s, t. Daraus folgt: $y = spy + txy$. Nun ist xy durch p teilbar, also auch $y = spy + txy$. □

Satz 4.28 ***(Fundamentalsatz der Arithmetik)*** **(∗)**
Jede ganze Zahl $x \geq 1$ kann als Produkt von Primzahlen geschrieben werden. Die Faktoren sind eindeutig bestimmt (bis auf die Reihenfolge).

Mit Hilfe von Lemma 4.24 können wir nun eine Formel für $\varphi(m) = |\mathbb{Z}_m^*|$ angeben, die auf der Primzahlzerlegung beruht.

Lemma 4.29
Für $m \geq 2$ gilt

$$\varphi(m) = m \cdot \prod_{\substack{p \text{ prim} \\ p \mid m}} \left(1 - \frac{1}{p}\right).$$

Beweis: Wenn m eine Primzahlpotenz p^t ist, dann besteht $\mathbb{Z}_m^*$ aus den Zahlen in $\mathbb{Z}_m = \{0, 1, \ldots, p^t - 1\}$, die nicht durch p teilbar sind. Da es in $\mathbb{Z}_m$ insgesamt p^t Zahlen gibt und p^{t-1} Vielfache von p, gilt $\varphi(m) = p^t - p^{t-1} = m - m/p = m(1 - 1/p)$. Nun nehmen wir an, dass $m = p_1^{t_1} \cdots p_s^{t_s}$ gilt, für verschiedene Primzahlen $p_1, \ldots, p_s$ und $t_1, \ldots, t_s \geq 1$. Die Faktoren $p_1^{t_1}, \ldots, p_s^{t_s}$ sind teilerfremd. denn wenn etwa $p_1^{t_1}$ und $p_2^{t_2} \ldots p_s^{t_s}$ einen gemeinsamen Teiler > 1 hätten, dann wäre p_1 Teiler von $p_2^{t_2} \ldots p_s^{t_s}$, was sofort einen Widerspruch zur Eindeutigkeit der Primfaktorzerlegung ergibt. Mit Lemma 4.24, $(s-1)$-mal angewendet, erhalten wir

$$\varphi(m) = \prod_{i=1}^{s} \varphi(p_i^{t_i}) = \prod_{i=1}^{s} (p_i^{t_i}(1 - 1/p_i)) = m \cdot \prod_{i=1}^{s} \left(1 - \frac{1}{p_i}\right).$$

□

Mit dieser Formel lassen sich die Werte in Tabelle 2 schnell verifizieren. (*Beispiel*: $\varphi(12) = 12(1-1/2)(1-1/3) = 12 \cdot (1/2) \cdot (2/3) = 4$.) Man beachte als Spezialfall: Wenn $m = pq$ für verschiedene Primzahlen p und q, dann ist $\varphi(m) = pq(1-1/p)(1-1/q) = (p-1)(q-1)$. (*Beispiel*: $\varphi(15) = 2 \cdot 4 = 8$.)

Bemerkung: Die einfache Formel in Lemma 4.29 könnte zu dem Schluss verleiten, dass sich $\varphi(m)$ zu gegebenem m immer leicht berechnen lässt. Aber Achtung: Man muss dazu die Menge der Primfaktoren von m kennen. Dies läuft darauf hinaus, das Faktorisierungsproblem für m zu lösen, also einen beliebigen Primfaktor für m zu finden, und hierfür kennt man keine effizienten Algorithmen. *Tatsächlich ist auch kein effizienter Algorithmus bekannt, der es erlaubt, $\varphi(m)$ aus m zu berechnen.*

Fakt 4.30
Jede zusammengesetzte Zahl x besitzt einen Primfaktor p mit $p \leq \sqrt{x}$.

Beweis: Man schreibt $x = yz$ für Zahlen y und z, die weder 1 noch x sind. Es ist nicht möglich, dass beide größer als $\sqrt{x}$ sind. Der kleinere Faktor enthält also einen Primfaktor, der nicht größer als $\sqrt{x}$ ist. □

Bemerkung: Wir betrachten das resultierende naive Faktorisierungsverfahren: Teste die Zahlen in $\{2, \ldots, \lfloor\sqrt{x}\rfloor\}$ nacheinander darauf, ob sie x teilen; wenn ein Faktor p gefunden wurde, wende dasselbe Verfahren auf $x' = x/p$ an. Dieses Verfahren hat im schlechtesten Fall Rechenzeit mindestens $\Theta(\sqrt{x}) = \Theta(2^{(\log x)/2})$, also exponentiell in der Bitlänge von x. Wie wir später genauer diskutieren werden, sind für das Auffinden der Primzahlzerlegung einer gegebenen Zahl x überhaupt keine effizienten Algorithmen bekannt (also Algorithmen mit Laufzeiten $O((\log x)^c)$ für konstantes c). Aber

es gibt effiziente Algorithmen, mit denen man feststellen kann, ob eine Zahl x eine Primzahl ist oder nicht. Dieser Unterschied in der Schwierigkeit des Faktorisierungsproblems und des Primzahlproblems liegt einer ganzen Reihe von kryptographischen Verfahren zugrunde.

Satz 4.31 ***(Euklid)***
Es gibt unendlich viele Primzahlen.

Beweis: Wenn $\{p_1, \ldots, p_k\}$, für $k \geq 1$, eine endliche Menge von (verschiedenen) Primzahlen ist, betrachten wir die Zahl $x = 1 + p_1 \cdots p_k$. Die Zahl x kann durch keine der Zahlen $p_1, \ldots, p_k$ teilbar sein, sonst wäre 1 durch diese Primzahl teilbar, was nicht möglich ist. Also sind alle Primfaktoren in der Primzahlzerlegung von x von $p_1, \ldots, p_k$ verschieden, es muss also außer $p_1, \ldots, p_k$ noch weitere Primzahlen geben. □

Über die Verteilung der Primzahlen (ihre „Dichte“) in $\mathbb{N}$ gibt der berühmte Primzahlsatz Auskunft.[8] Mit $\pi(x)$ bezeichnen wir die Anzahl der Primzahlen, die nicht größer als x sind.

Satz 4.32 ***Primzahlsatz***

$$\lim_{x\to\infty} \frac{\pi(x)}{x/\ln x} = 1.$$

Das heißt, dass für große x in $(x, 2x]$ etwa $\frac{2x}{\ln(2x)} - \frac{x}{\ln x} \approx \frac{x}{\ln x}$ Primzahlen zu erwarten sind. Die n-Bit-Zahlen bilden das Intervall $[2^{n-1}, 2n)$. Der Anteil der Primzahlen in diesem Intervall ist näherungsweise

$$\frac{2^{n-1}/\ln(2^{n-1})}{2^{n-1}} \approx \frac{1}{(\ln 2)(n-1)} \approx 1{,}44/n.$$

Für $n \approx 2000$ ist der relative Anteil von Primzahlen im interessanten Zahlenbereich also $\approx 1{,}44/2000 \approx 1/1400$. Er sinkt umgekehrt proportional zur Ziffernzahl.

Eine schärfere Form des Primzahlsatzes ist folgende Aussage[9] (wobei der Beweis Nichtspezialisten nicht zugänglich ist):

$$\frac{x}{\ln x}\left(1 + \frac{1}{\ln x}\right) \overset{(x\geq 599)}{\leq} \pi(x) \overset{(x\geq 2)}{\leq} \frac{x}{\ln x}\left(1 + \frac{1.2762}{\ln x}\right).$$

[8] 1793 von Gauß und unabhängig 1798 von Legendre vermutet; 1896 unabhängig von Hadamard und de La Vallée Poussin bewiesen.

[9] Corollary 5.2 in: Dusart, Pierre. Explicit estimates of some functions over primes. Ramanujan J. 45, 227–251 (2018). `https://doi.org/10.1007/s11139-016-9839-4`.

Für $x \geq 500\,000$ folgt daraus: $\frac{x}{\ln x} < \pi(x) < 1{,}1\frac{x}{\ln x}$.
Daraus folgt, dass für $n \geq 20$ der Anteil der Primzahlen unter den n-Bit-Zahlen folgende Ungleichung erfüllt:

$$\begin{aligned}\frac{|\{p \in [2^{n-1}, 2^n) \mid p \text{ is prime}\}|}{2^{n-1}} &= \frac{\pi(2^n) - \pi(2^{n-1})}{2^{n-1}} \\ &\geq \frac{2^n/(n\ln 2) - 1{,}1 \cdot 2^{n-1}/((n-1)\ln 2)}{2^{n-1}} \\ &\geq \frac{2}{n\ln 2} - \frac{1{,}1}{n\ln 2} \cdot \frac{n}{n-1} \\ &\geq \frac{6}{5n}.\end{aligned}$$

Mit Hilfe eines Computeralgebraprogramms findet man heraus, dass die Ungleichung $|\{p \in [2^{n-1}, 2^n) \mid p \text{ is prime}\}|/2^{n-1} \geq 6/(5n)$ auch für $9 \leq n \leq 20$ gilt. (Für $n = 8$ ist sie falsch.) Man kann sich also merken:

Für $n \geq 9$ ist der Anteil der Primzahlen an den n-Bit-Zahlen mindestens $\dfrac{6}{5n}$.

Ziffernzahl n	Dusart-Schranke für $(\pi(2^n) - \pi(2^{n-1}))/2^{n-1}$	numerische untere Schranke
256	$\frac{6}{5\cdot 256}$	$\geq \frac{1}{214}$
512	$\frac{6}{5\cdot 512}$	$\geq \frac{1}{427}$
1024	$\frac{6}{5\cdot 1024}$	$\geq \frac{1}{854}$
2048	$\frac{6}{5\cdot 2048}$	$\geq \frac{1}{1707}$

Eine leichter zu beweisende Aussage der Art $\pi(2m) - \pi(m) = \Theta(m/\log m)$ ist die folgende:

Satz 4.33 ***Ungleichung von Finsler***
Für jede ganze Zahl $m \geq 2$ liegen im Intervall $(m, 2m]$ mindestens $m/(3\ln(2m))$ Primzahlen*:*

$$\pi(2m) - \pi(m) \geq \frac{m}{3\ln(2m)}.$$

Ein vollständiger, vergleichsweise einfacher *Beweis* für Satz 4.33 findet sich zum Beispiel in dem Lehrbuch „Elemente der Diskreten Mathematik: Zahlen und Zählen, Graphen und Verbände“ von Diekert, Kufleitner, Rosenberger (De Gruyter 2013).

4.6 Der Primzahltest von Miller und Rabin

In diesem Abschnitt lernen wir einen randomisierten Algorithmus kennen, der es erlaubt, zu einer gegebenen Zahl N zu entscheiden, ob N eine Primzahl ist oder nicht.

Ein idealer Primzahltest sieht so aus:

Eingabe: Eine natürliche Zahl $N \geq 3$.

Ausgabe: 0, falls N eine Primzahl ist; 1, falls N zusammengesetzt ist.

Wozu braucht man Primzahltests? Zunächst ist die Frage „Ist N eine Primzahl?“ eine grundlegende mathematisch interessante Fragestellung. Spätestens mit dem Siegeszug des RSA-Kryptosystems (siehe Kapitel 5) hat sich die Situation jedoch dahin entwickelt, dass man Algorithmen benötigt, die immer wieder neue vielziffrige Primzahlen (etwa mit 1000 oder 1500 Bits[10] bzw. 301 oder 452 Dezimalziffern) bereitstellen können. Den Kern dieser Primzahlerzeugungs-Verfahren (siehe Abschnitt 4.7) bildet ein Verfahren, das eine gegebene Zahl N darauf testet, ob sie prim ist.

Ein naiver Primzahltest („versuchsweise Division“), der dem *brute-force*-Paradigma folgt, findet durch direkte Division der Zahl N durch $2, 3, 4, \ldots, \lfloor\sqrt{N}\rfloor$ heraus, ob N einen nichttrivialen Teiler hat. Man kann dieses Verfahren durch einige Tricks beschleunigen, aber die Rechenzeit wächst dennoch mit $\Theta(\sqrt{N})$. Dies macht es für Zahlen mit mehr als 40 Dezimalstellen praktisch undurchführbar, von Zahlen mit mehr als 100 Dezimalstellen ganz zu schweigen. (Achtung: Damit wird nichts über den Zeitaufwand bei anderen Faktorisierungsalgorithmen gesagt. Es gibt andere, sehr fortgeschrittene Faktorisierungsalgorithmen, die bei entsprechendem Zeitaufwand und mit sehr leistungsstarken Rechnern auch noch mit 200-stelligen Zahlen zurechtkommen. Für Information zu früheren und aktuelleren Faktorisierungserfolgen siehe z. B. `http://en.wikipedia.org/wiki/RSA_numbers`.)

In diesem Abschnitt beschreiben wir den randomisierten Primzahltest von Miller-Rabin. Dabei handelt es sich um einen „Monte-Carlo-Algorithmus mit einseitigem Fehler“. Das heißt: Auf Eingaben N, die Primzahlen sind, wird immer 0 ausgegeben;

[10] `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf` BSI TR-02102-1 „Kryptographische Verfahren: Empfehlungen und Schlüssellängen“, Version: 2020-01, S. 28: „Für einen Einsatzzeitraum über das Jahr 2022 hinaus wird empfohlen, RSA/DLIES-Schlüssel von 3000 Bits Länge zu verwenden, um ein gleichmäßiges Sicherheitsniveau in allen empfohlenen asymmetrischen Verschlüsselungsverfahren zu erzielen. Die Schlüssellänge von 2000 Bit bleibt voraussichtlich bis 2022 zur vorliegenden Richtlinie konform.“ Dabei bedeutet *Schlüssellänge* die Bitanzahl eines Produkts $n = p \cdot q$ für Primzahlen p und q mit jeweils der halben Länge.

auf Eingaben N, die zusammengesetzt sind, gibt es eine gewisse (von N abhängige) Wahrscheinlichkeit, dass die Ausgabe 0, also falsch ist. Für kein zusammengesetztes N ist diese Wahrscheinlichkeit größer als die „Fehlerschranke“ $\frac{1}{4}$. Wir beweisen nur die Fehlerschranke $\frac{1}{2}$. Im Beweis benutzen wir einfache zahlentheoretische Überlegungen. Eine herausragende Eigenschaft des Miller-Rabin-Tests ist seine Effizienz. Wir werden sehen, dass selbst bei Verwendung der Schulmethoden für Multiplikation und Division die Anzahl der Ziffernoperationen des Primzahltests nur $O((\log N)^3)$ ist.

Bemerkung: Der Miller-Rabin-Algorithmus stammt aus dem Jahr 1977; er folgte einem kurz vorher vorgestellten anderen randomisierten Primzahltest (Solovay-Strassen-Test). Für diesen und andere randomisierte Primzahltests (z. B. der „Strong Lucas Probable Prime Test“ oder der „Quadratic Frobenius Test“ von Grantham) sei auf die Literatur verwiesen. Im Jahr 2002 stellten Agarwal, Kayal und Saxena einen deterministischen Primzahltest mit polynomieller Rechenzeit vor. (Die Rechenzeit ist z. B. durch $O((\log N)^{7,5})$ beschränkt.) Dieser Algorithmus stellte insofern einen gewaltigen Durchbruch dar, als er ein Jahrhunderte altes offenes Problem löste, nämlich die Frage nach einem effizienten *deterministischen* Verfahren für das Entscheidungsproblem „ist N Primzahl oder zusammengesetzt“? Andererseits ist seine Laufzeit im Vergleich etwa zu dem hier diskutierten randomisierten Verfahren so hoch, dass nach wie vor die randomisierten Algorithmen benutzt werden, um für kryptographische Anwendungen Primzahlen zu erzeugen.

Da gerade Zahlen leicht zu erkennen sind, beschränken wir im Folgenden unsere Überlegungen auf ungerade Zahlen $N \geq 3$.

4.6.1 Der Fermat-Test

Wir erinnern uns an Fakt 4.20, den Kleinen Satz von Fermat: Wenn p eine Primzahl ist und $1 \leq a < p$, dann gilt $a^{p-1} \bmod p = 1$.

Wir können diese Aussage dazu benutzen, um „Belege“ oder „Zertifikate“ oder „Zeugen“ dafür anzugeben, dass eine Zahl N zusammengesetzt ist: Wenn wir eine Zahl a mit $1 \leq a < N$ finden, für die $a^{N-1} \bmod N \neq 1$ gilt, dann ist N definitiv keine Primzahl.

Beispiel: Mit $N = 15$ und $a = 2$ rechnen wir: $2^{14} \equiv (2^4)^3 \cdot 2^2 \equiv 16^3 \cdot 4 \equiv 1 \cdot 4 \equiv 4 \pmod{15}$. Also ist $2^{14} \bmod 15 = 4 \neq 1$, also ist 15 definitiv keine Primzahl. (Man beachte, dass wir keinen Faktor angeben müssen, um zu diesem Schluss zu kommen.)

Definition 4.34
Sei $N \geq 3$ ungerade und zusammengesetzt.
Eine Zahl $a \in \{1, \ldots, N-1\}$ heißt **F-Zeuge** für N, wenn $a^{N-1} \bmod N \neq 1$ gilt.
Eine Zahl $a \in \{1, \ldots, N-1\}$ heißt **F-Lügner** für N, wenn $a^{N-1} \bmod N = 1$ gilt.
Die Menge der F-Lügner nennen wir L_N^{F}.

Wir bemerken, dass ein F-Zeuge belegt, dass es Faktoren $k, \ell > 1$ mit $N = k \cdot \ell$ *gibt*, dass aber ein F-Zeuge nicht auf solche Faktoren hinweist oder sie beinhaltet. Das Finden von Faktoren wird von Primzahltests auch nicht verlangt und normalerweise auch nicht geleistet.

Man sieht sofort, dass 1 und $N-1$ immer F-Lügner sind: Es gilt $1^{N-1} \mod N = 1$ und $(N-1)^{N-1} \equiv (-1)^{N-1} = 1 \pmod N$, weil $N-1$ gerade ist.

Für jede zusammengesetzte Zahl N gibt es mindestens einen F-Zeugen. Nach Fakt 4.19 gilt $\{1, \ldots, N-1\} - \mathbb{Z}_N^* \neq \emptyset$, wenn N zusammengesetzt ist.

Lemma 4.35
Wenn N zusammengesetzt ist, ist jedes $a \in \{1, \ldots, N-1\} - \mathbb{Z}_N^*$ ein F-Zeuge.

Beweis: Sei $d = \mathrm{ggT}(a, N) > 1$. Dann ist auch a^{N-1} durch d teilbar, also auch $a^{N-1} \bmod N = a^{N-1} - \lfloor a^{N-1}/N \rfloor \cdot N$. Daher ist $a^{N-1} \bmod N \neq 1$. □

Beispiel: Für $N = 15$ und $a = 6$ gilt $6^{14} \equiv 36^7 \equiv 6^7 \equiv 36^3 \cdot 6 \equiv 6^4 \equiv 6^2 \equiv 6 \pmod{15}$. Der Rest 6 ist durch $\mathrm{ggT}(6, 15) = 3$ teilbar.

Leider ist für manche zusammengesetzten Zahlen N die Menge $\{1, \ldots, N-1\} - \mathbb{Z}_N^*$ äußerst dünn. Wenn zum Beispiel $N = pq$ für zwei Primzahlen p und q ist, dann gilt $\mathrm{ggT}(a, N) > 1$ genau dann wenn p oder q ein Teiler von a ist. Es gibt genau $p + q - 2$ solche Zahlen a in $\{1, \ldots, N-1\}$, was gegenüber N sehr klein ist, wenn p und q annähernd gleich groß sind. Um eine gute Chance zu haben, F-Zeugen zu finden, sollte es also mehr als nur die in $\{1, \ldots, N-1\} - \mathbb{Z}_N^*$ geben.

Beispiel: $N = 91 = 7 \cdot 13$. Tabelle 3 zeigt, dass es 18 Vielfache von 7 und 13 gibt (für größere p und q wird der Anteil dieser offensichtlichen F-Zeugen noch kleiner sein), und daneben weitere 36 F-Zeugen und 36 F-Lügner in $\{1, 2, \ldots, 90\}$.

In diesem Beispiel gibt es um einiges mehr F-Zeugen als F-Lügner. Wenn dies für alle zusammengesetzten Zahlen N der Fall wäre, wäre es eine elegante randomisierte Strategie, einfach zufällig nach F-Zeugen zu suchen.

Dies führt zu unserem ersten Versuch für einen randomisierten Primzahltest.

F-Zeugen in $\{1,\ldots,90\} - \mathbb{Z}_{91}^*$:	
7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84; 13, 26, 39, 52, 65, 78	
F-Lügner:	F-Zeugen in $\mathbb{Z}_{91}^*$:
1, 3, 4, 9, 10, 12, 16, 17, 22,	2, 5, 6, 8, 11, 15, 18, 19, 20,
23, 25, 27, 29, 30, 36, 38, 40, 43,	24, 31, 32, 33, 34, 37, 41, 44, 45,
48, 51, 53, 55, 61, 62, 64, 66, 68,	46, 47, 50, 54, 57, 58, 59, 60, 67,
69, 74, 75, 79, 81, 82, 87, 88, 90	71, 72, 73, 76, 80, 83, 85, 86, 89

Tabelle 3: F-Zeugen und F-Lügner für $N = 91 = 7 \cdot 13$. Es gibt 36 F-Lügner und 36 F-Zeugen in $\mathbb{Z}_{91}^*$. Wir wissen nach Lemma 4.35, dass alle 18 Vielfachen von 7 und 13 F-Zeugen sind.

Algorithmus 4.4 ***Fermat-Test***
EINGABE: Ungerade Zahl $N \geq 3$.
METHODE:
1 Wähle a zufällig aus $\{1,\ldots,N-1\}$;
2 **if** $a^{N-1} \bmod N \neq 1$ **then return** 1 **else return** 0.

Die Laufzeitanalyse liegt auf der Hand: Der teuerste Teil ist die Berechnung der Potenz $a^{N-1} \bmod N$ durch schnelle Exponentiation, die nach den Ergebnissen von Lemma 4.14 $O(\log N)$ arithmetische Operationen und $O((\log N)^3)$ Ziffernoperationen benötigt. Weiter ist es klar, dass der Algorithmus einen F-Zeugen gefunden hat, wenn er „1“ ausgibt, dass in diesem Fall also N zusammengesetzt sein muss. Umgekehrt ausgedrückt: Wenn N eine Primzahl ist, gibt der Fermat-Test garantiert „0“ aus.

Für $N = 91$ wird das falsche Ergebnis 0 ausgegeben, wenn als a einer der 36 F-Lügner gewählt wird. Die Wahrscheinlichkeit hierfür ist $\frac{36}{90} = \frac{2}{5} = 0{,}4$.

Für viele zusammengesetzte Zahlen N gibt es reichlich F-Zeugen, so dass der Fermat-Test für diese N mit konstanter Wahrscheinlichkeit das korrekte Ergebnis liefert. Wir analysieren das Verhalten des Fermat-Tests für solche „gutmütigen“ Eingabezahlen N (für die $N = 91$ ein typisches Beispiel ist).

Satz 4.36
Sei $N \geq 9$ eine ungerade zusammengesetzte Zahl. Wenn es mindestens einen F-Zeugen $b \in \mathbb{Z}_N^*$ gibt, dann liefert der Fermat-Test auf Eingabe N mit Wahrscheinlichkeit größer als $\frac{1}{2}$ die korrekte Antwort „1“.

Beweis: Sei $b \in \mathbb{Z}_N^*$ ein F-Zeuge. Betrachte die Funktion $g_b \colon L_N^{\mathrm{F}} \to \mathbb{Z}_N^*$, die den F-Lügner a auf $g_b(a) = ba \bmod N$ abbildet. Wie im Beweis von Fakt 4.20 sieht man, dass g_b injektiv ist. Weiter ist $g_b(a)$ für jedes $a \in L_N^{\mathrm{F}}$ ein F-Zeuge:

$$(ba \bmod N)^{N-1} \bmod N = (b^{N-1} \bmod N)(\underbrace{a^{N-1} \bmod N}_{=1}) = b^{N-1} \bmod N \neq 1.$$

Wir können also jedem F-Lügner a einen eigenen F-Zeugen $g_b(a)$ in $\mathbb{Z}_N^*$ zuordnen. Daraus folgt, dass es in $\mathbb{Z}_N^*$ mindestens so viele F-Zeugen wie F-Lügner gibt. Mit Lemma 4.35 ergibt sich, dass $\{1, \ldots, N-1\}$ mehr F-Zeugen als F-Lügner enthält. Daher ist die Wahrscheinlichkeit, dass die im Fermat-Test zufällig gewählte Zahl a ein F-Lügner ist, kleiner als $\frac{1}{2}$. □

Eine Fehlerwahrscheinlichkeit in der Nähe von $\frac{1}{2}$ ist natürlich viel zu groß. Wir verringern die Fehlerschranke durch wiederholte Ausführung des Fermat-Tests.

Algorithmus 4.5 ***Iterierter Fermat-Test***
EINGABE: Ungerade Zahl $N \geq 3$, eine Zahl $\ell \geq 1$.
METHODE:

```
1    repeat ℓ times
2        a ← ein zufälliges Element von {1, ..., N − 1};
3        if a^{N−1} mod N ≠ 1 then return 1;
4    return 0.
```

Wenn die Ausgabe 1 ist, hat der Algorithmus einen F-Zeugen für N gefunden, also ist N zusammengesetzt. D. h.: Wenn N eine Primzahl ist, ist die Ausgabe 0. Andererseits: Wenn N zusammengesetzt ist, und es mindestens einen F-Zeugen $b \in \mathbb{Z}_N^*$ gibt, dann ist nach Satz 4.36 die Wahrscheinlichkeit für die falsche Ausgabe „0“ höchstens $\left(\frac{1}{2}\right)^{\ell} = 2^{-\ell}$. Indem wir ℓ genügend groß wählen, können wir die Fehlerwahrscheinlichkeit so klein wie gewünscht einstellen.

Wenn es darum geht, aus einem genügend großen Bereich zufällig gewählte Zahlen darauf zu testen, ob es sich um eine Primzahl handelt, dann ist der Fermat-Test (in Kombination mit dem Testen auf kleine Teiler, etwa alle Primzahlen unter 1000) allem Anschein nach eine sehr effiziente und zuverlässige Methode. Dies wird durch

empirische Resultate nahegelegt.Wenn man allerdings über die Herkunft der zu testenden Zahl N keine Information hat und eventuell damit rechnen muss, dass jemand (ein „Gegenspieler“) absichtlich eine besonders schwierige Eingabe vorlegt, dann stößt der Fermat-Test an eine Grenze. Es gibt nämlich „widerspenstige“ zusammengesetzte Zahlen, denen man mit diesem Test nicht beikommen kann, weil alle Elemente von $\mathbb{Z}_N^*$ F-Lügner sind. Mit diesen befasst sich der folgende Abschnitt.

4.6.2 Carmichael-Zahlen

Definition 4.37
Eine ungerade zusammengesetzte Zahl N heißt eine **Carmichael-Zahl**, wenn für alle $a \in \mathbb{Z}_N^*$ die Gleichung $a^{N-1} \bmod N = 1$ gilt.

Die kleinste Carmichael-Zahl ist $561 = 3 \cdot 11 \cdot 17$. Weitere kleine Carmichael-Zahlen sind $1105 = 5 \cdot 13 \cdot 17$ und $1729 = 7 \cdot 13 \cdot 19$. Erst im Jahr 1994 wurde bewiesen, dass es unendlich viele Carmichael-Zahlen gibt, genauer: Wenn x genügend groß ist, dann gibt es in $\{N \in \mathbb{N} \mid N \leq x\}$ mehr als $x^{2/7}$ Carmichael-Zahlen. Die aktuell beste bekannte untere Schranke ist $x^{1/3}$. Von Erdős (1956) stammt die obere Schranke $x \cdot \exp(\frac{-c \ln x \ln\ln\ln x}{\ln\ln x})$, für eine Konstante $c > 0$, die zeigt, dass Carmichael-Zahlen *viel* seltener als Primzahlen sind.[11]

Wenn wir dem Fermat-Test eine Carmichael-Zahl N als Eingabe geben, ist die Wahrscheinlichkeit für die falsche Antwort 0 nach Lemma 4.29 genau

$$\frac{\varphi(N)}{N-1} > \frac{\varphi(N)}{N} = \prod_{\substack{p \text{ prim} \\ p \text{ teilt } N}} \left(1 - \frac{1}{p}\right) > 1 - \sum_{\substack{p \text{ prim} \\ p \text{ teilt } N}} \frac{1}{p}.$$

Diese Wahrscheinlichkeit liegt nahe an 1, wenn N nur wenige und relativ große Primfaktoren hat. An solchen Carmichael-Zahlen besteht etwa im Bereich der Zahlen im Bereich $[10^{17}, 10^{18}]$ kein Mangel, wie ein Blick in entsprechende Tabellen[12] zeigt. Zum Beispiel ist $N = 925619721362375041 = 425681 \cdot 1277041 \cdot 1702721$ eine 18-ziffrige Carmichael-Zahl mit $\varphi(N)/N > 0.999996$.

Der Wiederholungstrick zur Wahrscheinlichkeitsverbesserung hilft hier leider auch nicht, denn wenn etwa p_0 der kleinste Primfaktor von N ist, und N nur 3 oder 4

[11] Der Nächerungswert $\frac{x}{\ln x}$ für die Anzahl der Primzahlen $\leq x$ schreibt sich als $x \cdot \exp(-\ln\ln x)$, und $\ln\ln x \ll \frac{\ln x \ln\ln\ln x}{\ln\ln x}$ für $x \to \infty$. Mehr Details über Carmichaelzahlen findt man in Wikipedia: `https://en.wikipedia.org/wiki/Carmichael_number`.

[12] `http://www.s369624816.websitehome.co.uk/rgep/cartable.html`

Faktoren hat, dann sind $\Omega(p_0)$ Wiederholungen nötig, um die Fehlerwahrscheinlichkeit auf $\frac{1}{2}$ zu drücken. Sobald p_0 mehr als 30 Dezimalstellen hat, ist dies undurchführbar.

Für einen zuverlässigen, effizienten Primzahltest, der für *alle* Eingabezahlen funktioniert, müssen wir über den Fermat-Test hinausgehen. Interessanterweise ist dies praktisch ohne Effizienzverlust möglich.

Für spätere Benutzung stellen wir noch eine Hilfsaussage über Carmichael-Zahlen bereit.

Lemma 4.38
Wenn N eine Carmichael-Zahl ist, dann ist N keine Primzahlpotenz.

Beweis: Wir beweisen die Kontraposition: Wenn $N = p^\ell$ für eine ungerade Primzahl p und einen Exponenten $\ell \geq 2$ ist, dann ist N keine Carmichael-Zahl.

Dazu genügt es, eine Zahl $a \in \mathbb{Z}_N^*$ anzugeben, so dass $a^{N-1} \bmod N \neq 1$ ist. Wir definieren:

$$a := p^{\ell-1} + 1.$$

(Wenn z. B. $p = 7$ und $\ell = 3$ ist, ist $N = 343$ und $a = 49 + 1 = 50$.) Man sieht sofort, dass $a < p^\ell = N$ ist, und dass a nicht von p geteilt wird, also a und N teilerfremd sind; also ist $a \in \mathbb{Z}_N^*$. Nun rechnen wir modulo N, mit der binomischen Formel:

$$\begin{aligned} a^{N-1} &\equiv (p^{\ell-1} + 1)^{N-1} \\ &\equiv \sum_{0 \leq j \leq N-1} \binom{N-1}{j} (p^{\ell-1})^j \\ &\equiv 1 + (p^\ell - 1) \cdot p^{\ell-1} \pmod{N}. \end{aligned} \tag{4.3}$$

(Die letzte Äquivalenz ergibt sich daraus, dass für $j \geq 2$ gilt, dass $(\ell - 1)j \geq \ell$ ist, also der Faktor $(p^{\ell-1})^j = p^{(\ell-1)j}$ durch $N = p^\ell$ teilbar ist, also modulo N wegfällt.) Nun ist $p^\ell - 1$ nicht durch p teilbar, also ist $(p^\ell - 1) \cdot p^{\ell-1}$ nicht durch $N = p^\ell$ teilbar. Damit folgt aus (4.3), dass $a^{N-1} \not\equiv 1 \pmod{N}$ ist, also $a^{N-1} \bmod N \neq 1$. □

Folgerung: Jede Carmichael-Zahl N lässt sich als $N = N_1 \cdot N_2$ schreiben, wo N_1 und N_2 teilerfremde ungerade Zahlen ≥ 3 sind.

(Eine etwas genauere Untersuchung, die wir hier aber nicht benötigen, ergibt, dass die Primfaktoren einer Carmichael-Zahl N alle verschieden sein müssen, und dass N mindestens drei Primfaktoren haben muss. Auch aus dieser Tatsache kann man entnehmen, dass Carmichael-Zahlen eher selten sind.)

4.6.3 Nichttriviale Quadratwurzeln der 1

Beispiel: Betrachte $N = 7 \cdot 13$. Es gilt $1^2 = 1$ und $90^2 \equiv (-1)^2 \pmod{91}$. Aber es gilt auch $27^2 = 81 \cdot 9 \equiv (-10) \cdot 9 = -90 \equiv 1 \pmod{91}$. Daraus folgt auch $64^2 = (91 - 27)^2 \equiv (-27)^2 = 27^2 \equiv 1 \pmod{91}$.

Wir nennen eine Zahl $b \in \{2, \ldots, N-2\}$ mit $b^2 \bmod N = 1$ eine ***nichttriviale Quadratwurzel der* 1 modulo N**. Bei Primzahlen gibt es solche Zahlen nicht.

Lemma 4.39
Wenn p eine ungerade Primzahl ist, dann gilt $b^2 \bmod p = 1$, $b \in \mathbb{Z}_p$, genau für $b \in \{1, p-1\}$.

Beweis: Offensichtlich gilt für jedes beliebige $m \geq 2$, dass $1^2 \bmod m = 1$ und $(m-1)^2 \bmod m = (m(m-2)+1) \bmod m = 1$ ist. Nun sei $b \in \{0, \ldots, p-1\}$ beliebig mit $b^2 \equiv 1 \pmod{p}$. Dann gilt $b^2 - 1 \equiv 0 \pmod{p}$, also ist p ein Teiler von $b^2 - 1 = (b+1)(b-1)$. Nach Fakt 4.27 ist p Teiler von $b+1$ oder von $b-1$. Im ersten Fall ist $b \equiv -1 \pmod{p}$, im zweiten Fall ist $b \equiv 1 \pmod{p}$. □

Die im Lemma angegebene Eigenschaft lässt sich also in ein weiteres Zertifikat für zusammengesetzte Zahlen ummünzen:

> Wenn es eine nichttriviale Quadratwurzel der 1 modulo N gibt,
> dann ist N zusammengesetzt.

Die vier Zahlen 1, 27, 64 und 90 sind genau die Quadratwurzeln der 1 modulo 91; davon sind 27 und $64 = 91 - 27$ nichttrivial. Beachte, dass $1 \equiv 63 \pmod{7}$ und $27 \equiv 90 \equiv -1 \pmod{7}$, und dass $1 \equiv 27 \pmod{13}$ und $64 \equiv 90 \equiv -1 \pmod{13}$. Allgemeiner sieht man mit der Verallgemeinerung von Fakt 4.21 (Chinesischer Restsatz) auf r Faktoren leicht ein, dass es für ein Produkt $N = p_1 \cdots p_r$ aus verschiedenen ungeraden Primzahlen $p_1, \ldots, p_r$ genau 2^r Quadratwurzeln der 1 modulo N gibt, nämlich die Zahlen b, $0 \leq b < N$, die $b \bmod p_j \in \{1, p_j - 1\}$, $1 \leq j \leq r$, erfüllen. Wenn N nicht sehr viele verschiedene Primfaktoren hat, ist es also aussichtslos, einfach zufällig gewählte b's darauf zu testen, ob sie vielleicht nichttriviale Quadratwurzeln der 1 sind. Dennoch wird uns dieser Begriff bei der Formulierung eines effizienten Primzahltests helfen.

4.6.4 Der Miller-Rabin-Test

Wir kehren nochmals zum Fermat-Test zurück und sehen uns die dort durchgeführte Exponentiation $a^{N-1} \bmod N$ etwas genauer an. Die Zahl $N - 1$ ist gerade, daher

a	$b_0 = a^{81}$	$b_1 = a^{162}$	$b_2 = a^{324}$	F-Z.?	MR-Z.?	Fall
126	1	1	1			1
49	324	1	1			2a
7	307	324	1			2b
2	252	129	66	×	×	3
15	200	25	300	×	×	3
224	274	1	1		×	4a
201	226	51	1		×	4b

Tabelle 4: Potenzen $a^{N-1} \bmod N$ mit Zwischenschritten, $N = 325$. (Die „Fälle“ werden weiter unten erklärt.

kann man sie als $N - 1 = u \cdot 2^k$ schreiben, für eine ungerade Zahl u und ein $k \geq 1$. Dann gilt $a^{N-1} \equiv (a^u \bmod N)^{2^k} \bmod N$, und wir können $a^{N-1} \bmod N$ mit $k + 1$ Zwischenschritten berechnen: Mit

$$\begin{aligned}
b_0 &= a^u \bmod N \\
b_1 &= b_0^2 \bmod N = a^{u\cdot 2} \bmod N, \\
b_2 &= b_1^2 \bmod N = a^{u\cdot 2^2} \bmod N, \\
&\vdots \\
b_i &= b_{i-1}^2 \bmod N = a^{u\cdot 2^i} \bmod N, \\
&\vdots \\
b_k &= b_{k-1}^2 \bmod N = a^{u\cdot 2^k} \bmod N
\end{aligned}$$

ist $b_k = a^{N-1} \bmod N$. Beispielsweise erhalten wir für $N = 325 = 5^2 \cdot 13$ den Wert $N - 1 = 324 = 81 \cdot 2^2$. In Tabelle 4 berechnen wir a^{81}, a^{162} und a^{324}, alle modulo 325, für verschiedene a.

Die Grundidee des Miller-Rabin-Tests ist nun, diese verlangsamte Berechnung der Potenz $a^{N-1} \bmod N$ auszuführen und dabei nach nichttrivialen Quadratwurzeln der 1 Ausschau zu halten.

Im Beispiel sehen wir, dass 2 ein F-Zeuge für 325 ist, der in $\mathbb{Z}_{325}^*$ liegt, und dass 15 ein F-Zeuge ist, der nicht in $\mathbb{Z}_{325}^*$ liegt. Dagegen sind 126, 49, 7, 224 und 201 F-Lügner für 325. Wenn wir aber 224^{324} mod 325 mit zwei Zwischenschritten berechnen, dann entdecken wir, dass 274 eine nichttriviale Quadratwurzel der 1 ist, was beweist, dass 325 keine Primzahl ist. Ähnlich liefert die Berechnung mit Basis 201, dass 51 eine nichttriviale Quadratwurzel der 1 ist. Die entsprechenden Berechnungen mit 49 und 7 dagegen liefern keine weiteren Informationen, weil $49^{81} \equiv -1 \pmod{325}$ und $7^{162} \equiv 32^{162} \equiv -1 \pmod{325}$ gilt. Auch die Berechnung der Potenzen von 126 liefert keine nichttriviale Quadratwurzel der 1, weil $126^{81} \bmod 325 = 1$ gilt.

Wie kann die Folge $b_0, \ldots, b_k$ überhaupt aussehen? Wenn $b_i = 1$ or $b_i = N-1$ gilt, dann sind $b_{i+1}, \ldots, b_k$ alle gleich 1. Daher beginnt die Folge $b_0, \ldots, b_k$ mit einer (eventuell leeren) Folge von Elementen $\notin \{1, N-1\}$ und endet mit einer (eventuell leeren) Folge von Einsen. Diese beiden Teile können von einem Eintrag $N-1$ getrennt sein oder nicht.

Die möglichen Muster sind in Tabelle 5 angegeben. Dabei steht „$*$“ für ein Element $\notin \{1, N-1\}$. Wir unterscheiden vier Fälle (mit Unterfällen).

Fall 1: $b_0 = 1$. – Dann ist $b_1 = \cdots = b_k = 1$.

Fall 2: Die Folge endet mit $b_k = 1$, und vor der ersten 1 in der Folge steht eine $N-1$. (Fall 2a: $b_0 = N-1$; Fall 2b: $b_i = N-1$ für ein $i \in \{1, \ldots, k-1\}$.)

Fall 3: Die Folge endet mit $b_k \neq 1$. – Dann ist N zusammengesetzt, weil a ein F-Zeuge für N ist. In $b_0, \ldots, b_{k-1}$ können 1 und $N-1$ nicht vorkommen.

Fall 4: Die Folge endet mit $b_k = 1$, aber $b_0 \neq 1$ und in $b_0, \ldots, b_{k-1}$ kommt $N-1$ nicht vor. – Dann gibt es ein $i \in \{1, \ldots, k\}$ mit $b_{i-1} \notin \{1, N-1\}$ und $b_i = 1$. Das heißt, dass b_{i-1} eine nichttriviale Quadratwurzel der 1 ist. Also ist N zusammengesetzt. (Fall 4a: $i < k$; Fall 4b: $i = k$.)

In Tabelle 4 findet man konkrete Beispiele für das Eintreten aller Fälle für die zusammengesetzte Zahl $N = 325$.

Wir beobachten: Wenn N eine Primzahl ist, dann gilt nach dem kleinen Satz von Fermat für jedes a die Gleichung $b_k = a^{N-1} \bmod N = 1$. Weiter kann es nach Lemma 4.39 nicht passieren, dass $b_{i-1} \neq N-1$ und $b_i = 1$ ist. Also können für eine Primzahl N nur Fälle 1 und 2 vorkommen. Umgekehrt findet man im Beispiel $N = 17$, dass $a = 1$ zu Fall 1, $a = 16$ zu Fall 2a, und $a = 2$ bzw. $a = 3$ zu Fall 2b mit $b_2 = 16$ bzw. $b_3 = 16$ führt. Um bei Primzahlen keine falschen Ergebnisse zu erzeugen, müssen wir also in

b_0	b_1	$\cdots$				$\cdots$	b_{k-1}	b_k	Fall	F-Z.?	MR-Z.?
1	1	$\cdots$	1	1	1	$\cdots$	1	1	1		
$N-1$	1	$\cdots$	1	1	1	$\cdots$	1	1	2a		
$*$	$*$	$\cdots$	$*$	$N-1$	1	$\cdots$	1	1	2b		
$*$	$*$	$\cdots$	$*$	$*$	$*$	$\cdots$	$*$	$\neq 1$	3	$\times$	$\times$
$*$	$*$	$\cdots$	$*$	1	1	$\cdots$	1	1	4a		$\times$
$*$	$*$	$\cdots$	$*$	$*$	$*$	$\cdots$	$*$	1	4b		$\times$

Tabelle 5: Potenzen a^{N-1} mod N berechnet mit Zwischenschritten, mögliche Fälle.

den Fällen 1 und 2 den Wert 0 ausgeben. In den Fällen 3 und 4 hingegen stellt die Zahl a (mit ihren Potenzen $b_0, \ldots, b_k$) einen Beleg dafür dar, dass N keine Primzahl ist: Im Fall 3 ist a ein F-Zeuge, im Fall 4 ist b_{i-1} eine nichttriviale Quadratwurzel der 1, und das kann nur passieren, wenn N zusammengesetzt ist. Hier können wir also 1 ausgeben.

Das ist auch schon der ganze Algorithmus von Miller und Rabin, abstrakt formuliert: Wähle a aus $\{1, \ldots, N-1\}$ zufällig. Finde heraus, ob Fall 1 oder 2 eintritt (dann ist die Ausgabe 0) oder Fall 3 oder 4 eintritt (dann ist die Ausgabe 1).

Betrachten von Tab. 5 zeigt sofort, dass Fall 1 oder 2 genau dann eintritt, wenn Folgendes gilt:

$$b_0 = 1 \qquad \text{oder} \qquad \text{in der Folge } b_0, \ldots, b_{k-1} \text{ zu } a \text{ kommt } N-1 \text{ vor.} \tag{4.4}$$

(Das letzte Folgenglied b_k spielt keine Rolle für die Unterscheidung.) Dies führt zu folgender Definition in Analogie zu der der F-Zeugen und F-Lügner.

Definition 4.40
Sei $N \geq 3$ ungerade und zusammengesetzt. Wir schreiben $N - 1 = u \cdot 2^k$, für u ungerade, $k \geq 1$.
Eine Zahl a, $1 \leq a < N$, heißt ein **MR-Zeuge für N**, wenn (4.4) nicht gilt, d. h. $a^u \not\equiv 1$ und $a^{u \cdot 2^i} \not\equiv N-1 \pmod N$ für alle i mit $0 \leq i < k$ (Fälle 3 und 4).
Eine Zahl a, $1 \leq a < N$, heißt ein **MR-Lügner für N**, wenn (4.4) gilt, $a^u \equiv 1$ oder $a^{u \cdot 2^i} \equiv N-1 \pmod N$ für ein i mit $0 \leq i < k$ (Fälle 1 und 2).
Die Menge der MR-Lügner nennen wir L_N^{MR}.

Aus der obigen Diskussion der möglichen Fälle folgt sofort:

Lemma 4.41
Sei $N \geq 3$ ungerade.

(a) a ist F-Zeuge für $N \quad \Rightarrow \quad a$ ist MR-Zeuge für N (Fall 3).

(b) Wenn N eine Primzahl ist, dann gilt (4.4) für alle $a < N$.

Im Jahr 1976 stellte Gary M. Miller einen deterministischen Algorithmus vor, der auf der Idee beruhte, die Folge $b_0, \ldots, b_k$ zu betrachten. Er testete die kleinsten $O((\log N)^2)$ Elemente $a \in \{2, 3, \ldots, N-1\}$ auf die Eigenschaft, MR-Zeugen zu sein. Der Algorithmus hat polynomielle Laufzeit und liefert das korrekte Ergebnis, wenn die „erweiterte Riemannsche Vermutung (ERH)“ stimmt, eine berühmte Vermutung aus der Zahlentheorie, die bis heute unbewiesen ist. Um 1980 schlugen Michael Rabin und (unabhängig) Louis Monier vor, Millers Algorithmus so zu modifizieren, dass die zu testende Zahl a zufällig gewählt wird, und bewiesen, dass dieser Algorithmus konstante Fehlerwahrscheinlichkeit hat. Der Algorithmus heißt heute allgemein der ***Miller-Rabin-Test***.

Der Test selbst ist leicht und sehr effizient durchführbar: Man wählt a zufällig und berechnet zunächst $b_0 = a^u \bmod N$ und testet, ob $b_0 \in \{1, N-1\}$ ist. Falls ja, trifft Fall 1 bzw. Fall 2a zu, die Ausgabe ist 0. Falls nein, berechnet man durch iteriertes Quadrieren nacheinander die Werte $b_1, b_2, \ldots$, bis

- entweder der Wert $N-1$ auftaucht (Fall 2b, Ausgabe 0)
- oder der Wert 1 auftaucht (Fall 4a, a ist MR-Zeuge, Ausgabe 1)
- oder $b_1, \ldots, b_{k-1}$ berechnet worden sind, ohne dass Werte $N-1$ oder 1 vorgekommen sind (Fall 3, a ist F-Zeuge, oder 4b, a ist MR-Zeuge, Ausgabe 1).

In Pseudocode sieht das Verfahren dann folgendermaßen aus. Im Unterschied zur bisherigen Beschreibung wird nur eine Variable `b` benutzt, die nacheinander die Werte $b_0, b_1, \ldots$ aufnimmt.

Algorithmus 4.6 ***Der Miller-Rabin-Primzahltest***
EINGABE: Eine ungerade Zahl $N \geq 3$.
METHODE:

```
1    Bestimme u ungerade und k ≥ 1 mit N − 1 = u · 2^k;
2    wähle zufällig ein a aus {1, ..., N − 1};
3    b ← a^u mod N;                          // mit schnellem Potenzieren
4    if b ∈ {1, N − 1} then return 0;        // Fall 1 oder 2a
5    for j from 1 to k − 1 do                // „wiederhole (k − 1)-mal“
6        b ← b^2 mod N;
7        if b = N − 1 then return 0;         // Fall 2b
8        if b = 1 then return 1;             // Fall 4a
9    return 1.                               // Fall 3 oder 4b
```

Wie steht es mit der Rechenzeit des Algorithmus? Man benötigt höchstens $\log N$ Divisionen durch 2, um u und k zu finden (Zeile 1). Wenn man benutzt, dass N normalerweise in Binärdarstellung gegeben sein wird, ist die Sache noch einfacher: k ist die Zahl der Nullen, mit der die Binärdarstellung von $N-1$ aufhört, u ist die Zahl, die aus der Binärdarstellung von N durch Weglassen dieser letzten Nullen entsteht. Die Berechnung von a^u mod N mit schneller Exponentiation in Zeile 3 benötigt $O(\log N)$ arithmetische Operationen und $O((\log N)^3)$ Ziffernoperationen (mit der Schulmethode für Multiplikation und Division). Die Schleife in Zeilen 5–8 wird höchstens $(k-1)$-mal durchlaufen; offenbar ist $k \leq \log N$. In jedem Durchlauf ist die Multiplikation modulo N die teuerste Operation. Insgesamt benutzt der Algorithmus $O(\log N)$ arithmetische Operationen auf Zahlen, die kleiner als N^2 sind, und $O((\log N)^3)$ Ziffernoperationen. Wenn man die Rechenzeit mit der des Fermat-Tests vergleicht, stellt man fest, dass die Rechenwege etwas unterschiedlich sind,[13] aber dieselbe Anzahl von Multiplikationen modulo N anfallen. Nur werden beim Miller-Rabin-Test „unterwegs“ noch einige Zwischenergebnisse darauf getestet, ob sie gleich 1 oder gleich $N-1$ sind. Dies verursacht aber kaum Mehraufwand.

Nun wenden wir uns dem Ein-/Ausgabeverhalten des Miller-Rabin-Tests zu.

Lemma 4.42
Wenn N eine Primzahl ist, gibt der MR-Test 0 aus.

Beweis: Wir haben oben schon festgestellt, dass Ausgabe 1 genau dann erfolgt, wenn

[13] Es gibt einen alternativen Algorithmus für die schnelle Exponentiation, der exakt denselben Rechenweg verfolgt wie der Miller-Rabin-Algorithmus.

die zufällig gewählte Zahl a ein MR-Zeuge ist. Nach Lemma 4.41(b) gibt es nur dann MR-Zeugen, wenn N zusammengesetzt ist. □

4.6.5 Fehlerschranke für den Miller-Rabin-Test

Wir müssen nun noch die Fehlerwahrscheinlichkeit des Miller-Rabin-Tests für den Fall analysieren, dass N eine zusammengesetzte (ungerade) Zahl ist. Dies wird also für diesen Abschnitt angenommen.

Wir beweisen, dass die Wahrscheinlichkeit, dass der Miller-Rabin-Test die (unerwünschte) Antwort 0 gibt, kleiner als $\frac{1}{2}$ ist. Dazu wollen wir ähnlich vorgehen wie bei der Analyse des Fermat-Tests für Nicht-Carmichael-Zahlen (Beweis von Satz 4.36). Dort haben wir gezeigt, dass die F-Lügner für solche N höchstens die Hälfte von $\mathbb{Z}_N^*$ ausmachen.

Wenn N keine Carmichael-Zahl ist, ist dies leicht: Nach Lemma 4.41(a) gilt $L_N^{\mathrm{MR}} \subseteq L_N^{\mathrm{F}}$, und wir können den Beweis von Satz 4.36 verwenden.

Der folgende Beweis ist nicht prüfungsrelevant.

Es bleibt der Fall, dass N eine Carmichael-Zahl ist. Unser Plan ist, eine Menge B_N mit $L_N^{\mathrm{MR}} \subseteq B_N \subseteq \mathbb{Z}_N^*$ zu definieren, die höchstens die Hälfte der Elemente von $\mathbb{Z}_N^*$ enthält.

Weil u ungerade ist, gilt

$$(N-1)^{u\cdot 2^0} \equiv (N-1)^u \equiv (-1)^u = -1 \equiv N-1 \pmod{N}.$$

Sei i_0 das größte $i \in \{0,1,\ldots,k\}$ mit der Eigenschaft, dass es einen MR-Lügner $a_\#$ mit $a_\#^{u\cdot 2^i} \bmod N = N-1$ gibt. (Wir werden dieses $a_\#$ weiter unten nochmals benutzen.) Weil N eine Carmichael-Zahl ist, gilt $a^{u\cdot 2^k} \bmod N = a^{N-1} \bmod N = 1$ für alle $a \in \mathbb{Z}_N^*$, und daher $0 \le i_0 < k$. Wir definieren:

$$B_N := \{a \in \mathbb{Z}_N^* \mid a^{u\cdot 2^{i_0}} \bmod N \in \{1, N-1\}\}. \tag{4.5}$$

Zur Veranschaulichung betrachte man Tabelle 6. In der dort angegebenen Matrix entspricht jede Zeile einem der $\varphi(N)$ Elemente von $\mathbb{Z}_N^*$, beginnend mit $a_1 = 1$ und $a_2 = -1 \equiv N-1$. In der Zeile für a ist die von a erzeugte Folge $b_0, \ldots, b_k$ eingetragen. Dass N eine Carmichael-Zahl ist, drückt sich dadurch aus, dass alle Einträge in der b_k-Spalte gleich 1 sind. Der Eintrag für $a_2 = -1$ in der b_0-Spalte ist $N-1$. Spalte

a	b_0	b_1	$\cdots$		b_{i_0}		$\cdots$	b_{k-1}	b_k
$a_1 = 1$	1	1	$\cdots$	1	1	1	$\cdots$	1	1
$a_2 = N-1$	$N-1$	1	$\cdots$	1	1	1	$\cdots$	1	1
a_3	$*$	$*$	$\cdots$	$*$	$N-1$	1	$\cdots$	1	1
a_4	$*$	$*$	$\cdots$	$N-1$	1	1	$\cdots$	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	1	1
a	a^u	$a^{u\cdot 2^1}$	$\cdots$	$a^{u\cdot 2^{i_0-1}}$	$a^{u\cdot 2^{i_0}}$	$a^{u\cdot 2^{i_0+1}}$	$\cdots$	$a^{u\cdot 2^{k-1}}$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_\#$	$*$	$*$	$\cdots$	$*$	$N-1$	1	$\cdots$	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_?$	$*?$	$*?$	$\cdots$	$*?$	$*?$	$\cdots$	$\cdots$	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{\varphi(N)}$	?	?	$\cdots$	?	?	?	$\cdots$	?	1

Tabelle 6: Sicht auf B_N als Teilmenge von $\mathbb{Z}_N^*$

i_0 ist die am weitesten rechts stehende Spalte, in der es einen Eintrag $-1 \equiv N-1$ gibt, und $a_\#$ ist ein Element von $\mathbb{Z}_N^*$, das zu diesem Eintrag führt. Die Menge B_N besteht aus den Elementen von $\mathbb{Z}_N^*$, die in Spalte i_0 Eintrag 1 oder $N-1$ haben. – Wir werden zeigen, dass diese Menge B_N die gewünschten Eigenschaften hat.

Lemma 4.43

(a) $L_N^{\mathrm{MR}} \subseteq B_N$.

(b) $B_N \subsetneq \mathbb{Z}_N^*$.

(c) $|B_N| \leq \frac{1}{2}|\mathbb{Z}_N^*|$, also $\Pr_{a\in\{1,\ldots,N-1\}}(a \text{ ist MR-Lügner}) < \frac{1}{2}$.

Beweis: Teil (a) ist nur die Überprüfung, dass die Definition technisch sinnvoll ist. Der eigentlich interessante Teil ist (b). Teil (c) ist dann Routine.

(a) Sei a ein beliebiger MR-Lügner.
Fall 1: $a^u \bmod N = 1$. – Dann ist auch $a^{u\cdot 2^{i_0}} \bmod N = 1$, und daher gilt $a \in B_N$. (Die Zeile zu a in Abb. 6 sieht aus wie die von $a_1 = 1$.)
Fall 2: $a^{u\cdot 2^i} \bmod N = N-1$ für ein $i < k$. – Dann ist $0 \leq i \leq i_0$ nach der Definition von i_0. Wenn $i = i_0$ ist, haben wir direkt $a \in B_N$ (Beispiel in Tab. 6: a_3 oder $a_\#$); wenn $i < i_0$ ist, dann gilt

$$a^{u\cdot 2^{i_0}} \bmod N = (a^{u\cdot 2^i} \bmod N)^{2^{i_0-i}} \bmod N = (-1)^{2^{i_0-i}} \bmod N = 1,$$

also ebenfalls $a \in B_N$ (Beispiel in Tab. 6: a_2 oder a_4).

(b) Dass $B_N \subseteq \mathbb{Z}_N^*$ gilt, folgt aus der Defi nition. Wir müssen also nur zeigen, dass $\mathbb{Z}_N^* - B_N \neq \emptyset$ gilt. Wir benutzen die in Lemma 4.38 beobachtete Eigenschaft von Carmichael-Zahlen, keine Primzahlpotenz zu sein. Nach diesem Lemma können wir $N = N_1 \cdot N_2$ schreiben, für teilerfremde ungerade Zahlen $N_1, N_2 \geq 3$. Im Folgenden werden die Eigenschaften aus dem Chinesischen Restsatz (Fakt 4.21) benutzen.

Die grobe Idee der Konstruktion ist folgende: Wir haben das Element 1 mit $1^{u\cdot 2^{i_0}} \bmod N = 1$ und das Element $a_\#$ mit $a_\#^{u\cdot 2^{i_0}} \equiv -1 \pmod N$. Aus diesen beiden Elementen „basteln“ wir mit Hilfe des Chinesischen Restsatzes ein $b \in \mathbb{Z}_N^*$, das sich modulo N_1 wie 1 und modulo N_2 wie $a_\#$ verhält. Es wird sich zeigen, dass $b^{u\cdot 2^{i_0}} \bmod N \notin \{1, N-1\}$ gilt, also $b \notin B_N$ ist.

Wir führen diese Idee jetzt formal durch. Sei $x_2 = a_\# \bmod N_2$. Nach dem Chinesischen Restsatz (Fakt 4.21) gibt es eine eindeutig bestimmte Zahl $b \in \{0, 1, \ldots, N-1\}$, die

$$b \equiv 1 \pmod{N_1} \quad \text{und} \quad b \equiv x_2 \pmod{N_2} \tag{4.6}$$

erfüllt. (Es folgt $b \equiv a_\# \pmod{N_2}$.) Wir zeigen, dass b in $\mathbb{Z}_N^* - B_N$ liegt.
Wir notieren, dass für beliebige $x, y \in \mathbb{Z}$ gilt:

$$x \equiv x' \pmod{N} \qquad \Rightarrow \qquad x \equiv x' \pmod{N_i}, \text{ für } i = 1, 2. \tag{4.7}$$

(Wenn $x - x'$ durch N teilbar ist, dann auch durch N_1 und N_2.)

Beh. 1: $b \in \mathbb{Z}_N^*$.
Bew.: Offensichtlich gilt $b^{N-1} \equiv 1^{N-1} \equiv 1 \pmod{N_1}$. Weiter haben wir, modulo N_2 gerechnet:

$$b^{N-1} \equiv a_\#^{N-1} \overset{(4.7)}{\equiv} ((a_\#^{N-1}) \bmod N) \equiv 1 \pmod{N_2}.$$

Wegen der Eindeutigkeitsaussage im Chinesischen Restsatz folgt daraus $b^{N-1} \equiv 1 \pmod{N}$. Nach Lemma 4.35 folgt $b \in \mathbb{Z}_N^*$.

Beh. 2: $b \notin B_N$.
Bew.: Indirekt. Annahme: $b \in B_N$, d. h. $b^{u \cdot 2^{i_0}} \equiv 1 \pmod{N}$ oder $b^{u \cdot 2^{i_0}} \equiv -1 \pmod{N}$.
1. *Fall*: $b^{u \cdot 2^{i_0}} \equiv 1 \pmod{N}$. – Mit (4.7) folgt $b^{u \cdot 2^{i_0}} \equiv 1 \pmod{N_2}$. Andererseits gilt

$$b^{u \cdot 2^{i_0}} \equiv x_2^{u \cdot 2^{i_0}} \equiv a_\#^{u \cdot 2^{i_0}} \overset{(4.7)}{\equiv} (a_\#^{u \cdot 2^{i_0}} \bmod N) \equiv N - 1 \equiv -1 \pmod{N_2},$$

also $2 \equiv 0 \pmod{N_2}$, ein Widerspruch, weil $N_2 \geq 3$ ist.
2. *Fall*: $b^{u \cdot 2^{i_0}} \equiv -1 \pmod{N}$. – Mit (4.7) folgt $b^{u \cdot 2^{i_0}} \equiv -1 \pmod{N_1}$. Andererseits gilt

$$b^{u \cdot 2^{i_0}} \equiv 1^{u \cdot 2^{i_0}} \equiv 1 \pmod{N_1},$$

also $2 \equiv 0 \pmod{N_1}$, ebenfalls ein Widerspruch.

(c) Wir verwenden die in (b) konstruierte Zahl $b \in \mathbb{Z}_N^* - B_N$. Wie im Beweis von Satz 4.36 betrachtet man die injektive Funktion $g_b \colon B_N \ni a \mapsto ba \bmod N \in \mathbb{Z}_N^*$. Es gilt $g_b(a) \notin B_N$ für jedes $a \in B_N$, denn

$$g_b(a)^{u \cdot 2^{i_0}} \bmod N = (b^{u \cdot 2^{i_0}} \bmod N)(a^{u \cdot 2^{i_0}}) \bmod N \in \{b^{u \cdot 2^{i_0}} \bmod N, (N - b^{u \cdot 2^{i_0}}) \bmod N\},$$

und die letzte Menge ist disjunkt zu $\{1, N-1\}$. Man erhält sofort $|B_N| \leq \frac{1}{2}|\mathbb{Z}_N^*|$. □

Ab hier wieder prüfungsrelevant.

Wir haben also eine Schranke von $\frac{1}{2}$ für die Irrtumswahrscheinlichkeit im Miller-Rabin-Algorithmus bewiesen. Eine etwas kompliziertere Analyse zeigt, dass sogar die

Fehlerschranke $\frac{1}{4}$ gilt; man kann auch zeigen, dass es zusammengesetzte Zahlen N gibt (zum Beispiel $703 = 19 \cdot 37$, siehe eigenes Blatt), bei denen eine Fehlerwahrscheinlichkeit von fast $\frac{1}{4}$ tatsächlich auftritt. Durch ℓ-fache Wiederholung, ebenso wie in Algorithmus 4.5, kann man die Fehlerschranke auf $4^{-\ell}$ reduzieren. Wir kürzen den Miller-Rabin-Test mit „MiRa-Test“ ab. Man beachte, dass bei jedem neuen Aufruf eine neue Zufallszahl a gewählt wird.

Algorithmus 4.7 ***Iterierter MR-Test***
EINGABE: Ungerade Zahl $N \geq 3$, eine Zahl $\ell \geq 1$.
METHODE:
```
1     repeat ℓ times
2         if MiRa-Test(N) = 1 then return 1;
3     return 0;
```

Diesen Test wollen wir kurz IterMiRa(N, ℓ) nennen. – Wir erhalten zusammenfassend:

Proposition 4.44
Algorithmus 4.7 benötigt $O(\ell \cdot \log N)$ arithmetische Operationen auf Zahlen, die kleiner als N^2 sind, und $O(\ell \cdot (\log N)^3)$ Ziffernoperationen. Wenn N eine Primzahl ist, ist die Ausgabe 0, wenn N zusammengesetzt ist, ist die Wahrscheinlichkeit, dass 0 ausgegeben wird, kleiner als $4^{-\ell}$. □

Bemerkung: Auch in der Kryptographie kann man es mit Angreifern zu tun bekommen, die der Welt eine Nicht-Primzahl als Primzahl zu verkaufen versuchen. Situationen und Fallstricke bei Primzahltests für von Gegnern vorgelegte zusammengesetzte Zahlen werden in der Arbeit [Martin R. Albrecht, Jake Massimo, Kenneth G. Paterson, Juraj Somorovsky: Prime and Prejudice: Primality Testing Under Adversarial Conditions. CCS 2018: 281–298] erläutert.

4.7 Die Erzeugung von (zufälligen) Primzahlen

Für kryptographische Anwendungen (zum Beispiel für die Erzeugung von Schlüsselpaaren für das RSA-Public-Key-Kryptosystem) werden vielziffrige Primzahlen benötigt. Eine typische Aufgabe in diesem Zusammenhang lautet:

Finde eine zufällige Primzahl mit n Bits!

Das heißt: Gesucht ist eine zufällige Primzahl in $[2^{n-1}, 2^n)$. Dabei hängt n von der Anwendung ab; wir können uns z. B. $n = 1024$ oder $n = 2048$ vorstellen.

Man erinnere sich an den Primzahlsatz und an die Dusart-Schranke am Ende von Abschnitt 4.5, die besagt, dass es für $n \geq 9$ in diesem Intervall mindestens $(6/5) \cdot 2^{n-1}/n$ Primzahlen gibt.

Ein naheliegender Ansatz zur Erzeugung einer zufälligen n-Bit-Primzahl ist dann folgender: Man wählt wiederholt eine (ungerade) Zahl N aus $[2^{n-1}, 2^n)$ zufällig und wendet auf sie den (iterierten) Miller-Rabin-Test an. Dies wiederholt man, bis eine Zahl gefunden wurde, die die Ausgabe 0 liefert.

Algorithmus 4.8 ***GetPrime – Randomisierte Primzahlerzeugung***
EINGABE: Bitanzahl $n \geq 9$, Zahl $\ell \geq 1$ // Zuverlässigkeitsparameter
METHODE:
1 **repeat**
2 $\quad$ N $\leftarrow$ zufällige ungerade Zahl in $[2^{n-1}, 2^n)$;
3 **until** IterMiRa(N, ℓ) $= 0$; // Algorithmus 4.7
4 **return** N.

Die Ausgabe ist korrekt, wenn der Algorithmus eine Primzahl zurückgibt, ein Fehler tritt auf, wenn eine zusammengesetzte Zahl zurückgegeben wird. Auf den ersten Blick könnte man meinen (aufgrund von Proposition 4.44), dass die Fehlerwahrscheinlichkeit höchstens $1/4^\ell$ ist – eben die Fehlerwahrscheinlichkeit des iterierten MiRa-Tests. Wir werden sehen, dass wir auf der Basis der Analyse des Miller-Rabin-Tests nur ein etwas schwächeres Ergebnis bekommen.[14]

Wir definieren: $\text{Prim}_n := \{p \in [2^{n-1}, 2^n) \mid p \text{ ist Primzahl}\}$.

[14] Tatsächlich ist die Fehlerwahrscheinlichkeit durch $1/4^\ell$ beschränkt, für (von ℓ abhängig) genügend große n. Dies kann man aber nur durch fortgeschrittene zahlentheoretische Untersuchungen über die erwartete Wahrscheinlichkeit, dass eine zufällige ungerade zusammengesetzte Zahl den ℓ-fach iterierten MiRa-Test übersteht, beweisen [P. Beauchemin, G. Brassard, C. Crépeau, C. Goutier, C. Pomerance: The generation of random numbers that are probably prime. *J. Cryptology* **1**(1): 53-64 (1988)].

Satz 4.45
Bei der Anwendung von Algorithmus 4.8 auf $n \geq 9$ gilt*:*

(a) Wenn das Ergebnis eine Primzahl ist, hat jede Primzahl in $[2^{n-1}, 2^n)$ dieselbe Wahrscheinlichkeit, als Ergebnis zu erscheinen.

(b)
$$\Pr(\text{GetPrime}(n, \ell) \notin \text{Prim}_n) \leq \frac{5n}{12 \cdot 4^\ell} = O\left(\frac{n}{4^\ell}\right).$$
((!) Nicht $1/4^\ell$, wie naiv vermutet.)

(c) Die erwartete Rundenzahl ist $O(n)$, der erwartete Rechenaufwand ist $O(n^2)$ arithmetische Operationen und $O(n^4)$ Ziffernoperationen.

Beweis: (a) Eine Primzahl wird als Resultat genau dann geliefert, wenn keine zusammengesetzte Zahl fälschlicherweise vom Miller-Rabin-Test akzeptiert wird, bevor (in Zeile 2) die erste echte Primzahl gewählt wird. Jede der Primzahlen in $[2^{n-1}, 2^n)$ hat dieselbe Wahrscheinlichkeit, diese erste gewählte Primzahl zu sein.

(b)
$$\begin{aligned}
&\Pr(\text{GetPrime}(n, \ell) \notin \text{Prim}_n) \\
&= \Pr\big(\exists i \geq 1\colon \text{in Runden } j = 1, \ldots, i-1 \text{ wird eine} \\
&\qquad\qquad \text{zusammengesetzte Zahl } N \text{ gewählt } \textit{und erkannt } \wedge \\
&\qquad\qquad \text{in Runde } i \text{ wird zusammengesetzte Zahl } N \text{ gewählt} \\
&\qquad\qquad \text{und der iterierte MiRa-Test auf } N \text{ liefert } 0\big) \\
&\leq \sum_{i \geq 1} \left(1 - \frac{|\text{Prim}_n|}{2^{n-2}}\right)^{i-1} \cdot \frac{1}{4^\ell} \\
&= \frac{2^{n-2}}{|\text{Prim}_n|} \cdot \frac{1}{4^\ell} \\
&\leq \frac{2^{n-2}}{\frac{6}{5} \cdot 2^{n-1}/n} \cdot \frac{1}{4^\ell} \qquad \text{(wegen der Dusart-Schranke)} \\
&= \frac{5n}{12 \cdot 4^\ell}.
\end{aligned}$$

Wir haben benutzt, dass es in $[2^{n-1}, 2^n)$ genau 2^{n-2} ungerade Zahlen gibt.

(c) Da man in jeder Runde mit Wahrscheinlichkeit mindestens $\frac{|\text{Prim}_n|}{2^{n-2}}$ eine Primzahl wählt, ist die erwartete Rundenzahl nicht größer als $\frac{2^{n-2}}{|\text{Prim}_n|} \leq \frac{5n}{12}$. □

Bemerkung: Bei der Erzeugung zufälliger n-Bit-Primzahlen für kryptographische Zwecke wird man aus Effizienzgründen nicht unseren Algorithmus anwenden, der $\Theta(n^2)$ Multiplikationen benötigt, sondern eine Kombination zweier verschiedener Primzahltests (z. B. Miller-Rabin und „Lucas Strong Probable Prime Test“) mit sehr wenigen Iterationen und einem Test auf Teilbarkeit durch sehr kleine Primteiler. Dies erfordert nur $O(n)$ Multiplikationen. Die Dichte der zusammengesetzten Zahlen, die von einem solchen Test nicht erkannt werden, ist als sehr gering einzuschätzen. Über eine interessante experimentelle Untersuchung hierzu berichtet die kurze Notiz `http://people.csail.mit.edu/rivest/Rivest-FindingFourMillionLargeRandomPrimes.ps` von Ron Rivest (bekannt von „RSA“ und von „Cormen, Leiserson, Rivest und Stein“).

A Beweise und Bemerkungen für Abschnitt 4

Beweis der Existenz und der Eindeutigkeit des größten gemeinsamen Teilers $\mathrm{ggT}(x, y)$ zweier Zahlen (Definition 4.3(b)):

Eindeutigkeit: Wenn in (b) d und d' beide (i) und (ii) erfüllen und nichtnegativ sind, dann folgt $d \mid d'$ und $d' \mid d$, also $d = d'$ nach Fakt 4.2(e).

Existenz: Weil t gemeinsamer Teiler von x und y ist genau dann wenn t gemeinsamer Teiler von $a = |x|$ und $b = |y|$ ist, und weil offenbar das Vertauschen von x und y nichts ändert, können wir uns auf den Fall $x = a \geq y = b \geq 0$ beschränken. Wir zeigen durch Induktion über $b = \min\{a, b\}$, dass $\mathrm{ggT}(a, b)$ existiert.

Induktionsanfang: $b = 0$. *1. Fall*: $a = 0$. Dann ist jede Zahl t ein gemeinsamer Teiler von a und b. Wir wählen $d = 0$. Dann gilt (i), weil 0 Teiler von 0 ist, und (ii), weil jede Zahl t die Zahl 0 teilt. (0 ist „größter“ gemeinsamer Teiler von 0 und 0 im Sinn der Quasiordnung „Teilbarkeit“. Hier ist 0 das größte Element überhaupt.)
2. Fall: $a > 0$. Wir wählen $d = a$. Dann gilt (i), weil a Teiler von a und von 0 ist, und es gilt (ii), weil jeder gemeinsame Teiler von a und 0 auf jeden Fall Teiler von a ist.

Induktionsschritt: $b > 0$. Setze $q := a \operatorname{div} b$ und $r := a - qb = a \bmod b$ und $(a', b') := (b, r)$. Dann ist $b' = r < b = a'$. Nun haben a und b genau dieselben gemeinsamen Teiler wie a' und b'. (Aus $t \mid a$ und $t \mid b$ folgt $t \mid (a - qb)$, also $t \mid a'$, und aus $t \mid a'$ und $t \mid b'$ folgt $t \mid r + qb$, also $t \mid a$.) Nach I.V. existiert $d = \mathrm{ggT}(a', b')$, und dieses d ist dann auch größter gemeinsamer Teiler von a und b.

Bemerkung A.1 zu Fakt 4.12
Wenn man es genau nimmt, sind die Elemente von $\mathbb{Z}_m$ nicht Zahlen, sondern die Äquivalenzklassen („Restklassen“)

$$[x] = \{x + qm \mid q \in \mathbb{Z}\}, \text{ für } 0 \leq x < m,$$

zur Äquivalenzrelation $x \equiv y \pmod m$. Die Operationen werden auf den Repräsentanten ausgeführt, zum Beispiel gilt in $\mathbb{Z}_{11}$, dass $[4] + [7] = [11] = [0]$ ist, also sind $[4]$ und $[7]$ zueinander invers. Allgemeiner gilt in jedem $\mathbb{Z}_m$ die Gleichheit $[a] + [m - a] = [m] = [0]$, d. h., $[m - a]$ ist das additive Inverse $-[a]$ zu $[a]$. Multiplikativ gilt in $\mathbb{Z}_m$, dass $[1]$ neutrales Element ist, und dass $[1] \cdot [1] = [1]$ und $[m - 1] \cdot [m - 1] = [m(m - 2) + 1] = [1]$, also $[1]$ und $-[1] = [m - 1]$ (triviale) Quadratwurzeln der 1 sind.
Es ist bequem, die eckigen Klammern wegzulassen und für $[i]$ einfach i zu schreiben. Man muss dann nur aufpassen, dass man Ausdrücke wie „-1“ richtig interpretiert, nämlich als das additive Inverse von 1 in $\mathbb{Z}_m$, also $[m - 1]$.

Beweis von Satz 4.28:
Die Existenz der Zerlegung in Primfaktoren beweist man durch Induktion über x. Die Zahl $x = 1$ kann als leeres Produkt $\prod_{p \in \emptyset, p \text{ prim}} p$ geschrieben werden. Eine andere Darstellung gibt es nicht. Für $x > 1$ sucht man einen echten Faktor x_1 von x (d. h. $1 < x_1 < x$ und $x_1 y_1 = x$ für ein y_1), dann einen echten Faktor x_2 von x_1 usw., bis man einen echten Faktor x_k von x erhält, der selber keinen echten Faktor hat. Dieses x_k ist ein Primfaktor von x, man kann also $x = x' \cdot x_k$ schreiben. Auf x' wendet man die Induktionsvoraussetzung an.

Die Eindeutigkeit sieht man wie folgt mit einem indirekten Beweis: Annahme: Es gibt ein x mit unterschiedlichen Primfaktorzerlegungen. Wähle das kleinste solche x und unterschiedliche Primfaktorzerlegungen $p_1 \cdots p_k$ und $q_1 \cdots q_\ell$ von x. Aus der Minimalität von x folgt, dass in der Liste $p_1, \ldots, p_k$ keine der Primzahlen $q_1, \ldots, q_\ell$ vorkommt. Nun teilt q_1 die Zahl $x = p_1 \cdots p_k$. Nach Fakt 4.27, wiederholt angewendet, teilt q_1 einen der Faktoren $p_1, \ldots, p_k$, etwa p_i. Weil q_1 und p_i Primzahlen sind, muss $q_1 = p_i$ gelten, ein Widerspruch. □

Beweis von Fakt 4.16:
„⇒“: Es sei $ab \bmod m = 1$. Das heißt: Es gibt ein $q \in \mathbb{Z}$ mit $ab - qm = 1$. Wenn nun eine Zahl $d > 0$ sowohl a als auch m teilt, dann teilt d auch $ab - qm = 1$, also ist $d = 1$. Also sind a und m teilerfremd.

„⇐“: a und m seien teilerfremd. Nach Lemma 4.6(a) gibt es $x, y \in \mathbb{Z}$ mit $xa+ym = 1$. Setze $b := x \bmod m$. Dann gilt:

$$(a \cdot b) \bmod m = (a \cdot (x \bmod m)) \bmod m = ax \bmod m = 1,$$

d. h. b ist ein multiplikatives Inverses von a. □

Beweis von Fakt 4.18: Natürlich ist die 1 das neutrale Element der Multiplikation in $\mathbb{Z}_m$ und in $\mathbb{Z}_m^*$. Wir bemerken zuerst, dass $\mathbb{Z}_m^*$ unter Multiplikation modulo m abgeschlossen ist: Seien x und y zu m teilerfremd. Nach Lemma 4.6(a) kann man $1 = sx + tm = uy + vm$ schreiben, und erhält daraus

$$1 = (sx + tm)(uy + vm) = (su)(xy) + (sxv + tuy + tvm)m.$$

Daraus folgt, dass jeder gemeinsame Teiler von xy und m auch 1 teilen muss, also ist $\mathrm{ggT}(xy, m) = 1$.

Nach Fakt 4.16 hat jedes Element x von $\mathbb{Z}_m^*$ ein multiplikatives Inverses $y \in \mathbb{Z}_m$. Weil natürlich x auch das Inverse von y ist, folgt nach Fakt 4.16 auch, dass $y \in \mathbb{Z}_m^*$ ist.

Kommutativität ist klar. □

Beweis von Fakt 4.19: Die zweite Äquivalenz ist klar: Der Ring $\mathbb{Z}_m$ ist nach Definition ein Körper genau dann wenn jedes Element von $\mathbb{Z}_m - \{0\}$ ein multiplikatives Inverses besitzt.

Erste Äquivalenz:

„⇒“: Sei m eine Primzahl. Dann ist jedes $a \in \{1, \ldots, m-1\} = \mathbb{Z}_m - \{0\}$ zu m teilerfremd; nach der Definition folgt $\mathbb{Z}_m^* = \mathbb{Z}_m - \{0\}$.

„⇐“: Sei m eine zusammengesetzte Zahl, etwa durch k mit $2 \le k < m$ teilbar. Dann ist $k = \mathrm{ggT}(k, m) > 1$, also kann k nach Fakt 4.16 kein multiplikatives Inverses modulo m haben. (Man sieht auch direkt, dass $kb \bmod m = kb - qm$ durch k teilbar ist, also für kein b gleich 1 sein kann.) □

Beweis von Fakt 4.21: Wir beschränken uns darauf, die Bijektivität zu beweisen. Als erstes bemerkt man, dass $\mathbb{Z}_{mn}$ und $\mathbb{Z}_m \times \mathbb{Z}_n$ beide genau $m \cdot n$ Elemente haben. Daher reicht es zu zeigen, dass die Abbildung Φ injektiv ist. Seien dazu $0 \le x \le y < mn$ beliebig mit $\Phi(x) = \Phi(y)$. Das heißt: $x \bmod m = y \bmod m$ und $x \bmod N = y \bmod N$. Das wiederum heißt, dass $m \mid (y - x)$ und $n \mid (y - x)$ gilt. Nach Fakt 4.8 folgt, dass

mn Teiler von $y-x$ ist. Nun ist $0 \leq y-x < mn$, woraus $y-x=0$, also $x=y$, folgt. Damit ist die Injektivität von Φ bewiesen. □

Beweis von Prop. 4.22: Sei $x \in \mathbb{Z}_{mn}^*$. Dann gibt es in $\mathbb{Z}_{mn}^*$ ein Inverses y mit $xy \bmod mn = 1$. Daraus folgt $xy \bmod m = 1$, also ist $x \bmod m \in \mathbb{Z}_m^*$. Die entsprechende Aussage für n ergibt sich analog. – Umgekehrt müssen wir zeigen, dass für $u \in \mathbb{Z}_m^*$ und $v \in \mathbb{Z}_n^*$ die eindeutig bestimmte Zahl $x \in \mathbb{Z}_m$ mit $\Phi(x) = (x \bmod m, x \bmod n) = (u, v)$ teilerfremd zu mn ist. Wähle Inverse s von u in $\mathbb{Z}_m^*$ und t von v in $\mathbb{Z}_n^*$ und wähle nach dem Chinesischen Restsatz y mit $\Phi(y) = (s,t)$. Dann gilt $xy \bmod m = us \bmod m = 1$ und $xy \bmod n = vt \bmod n = 1$, also nach der Eindeutigkeitsaussage im Chinesischen Restsatz $xy \bmod mn = 1$, also $x \in \mathbb{Z}_{mn}^*$. □

Beweis von Satz 4.25:
Man geht genauso vor wie im Beweis des kleinen Satzes von Fermat. Sei x mit $\mathrm{ggT}(m,x) = 1$ gegeben. Da wir modulo m rechnen, können wir x durch $x \bmod m$ ersetzen, d. h., wir können annehmen, dass $1 \leq x < m$ gilt. Betrachte die Abbildung $g_x \colon \mathbb{Z}_m^* \ni s \mapsto xs \bmod m \in \mathbb{Z}_m^*$. (Um zu sehen, dass diese Abbildung wohldefiniert ist, erinnere man sich, dass mit x und s auch xs teilerfremd zu m ist.) Diese Abbildung ist injektiv, weil für das multiplikative Inverse y von x in $\mathbb{Z}_m^*$ die Gleichung $y \cdot g_x(s) \bmod m = yxs \bmod m = s$ gilt. Weil $\mathbb{Z}_m^*$ endlich ist, muss die Abbildung g_x sogar bijektiv sein. Das heißt: $\{g_x(s) \mid s \in \mathbb{Z}_m^*\} = \mathbb{Z}_m^*$. Daher:

$$\prod_{s \in \mathbb{Z}_m^*} s \equiv \prod_{s \in \mathbb{Z}_m^*} g_x(s) \equiv \prod_{s \in \mathbb{Z}_m^*} xs \equiv x^{\varphi(m)} \cdot \prod_{s \in \mathbb{Z}_m^*} s \pmod{m}.$$

Die Zahl $(\prod_{s \in \mathbb{Z}_m^*} s) \bmod m$ ist teilerfremd zu m, hat also ein multiplikatives Inverses z in $\mathbb{Z}_m^*$. Wenn wir beide Seiten der Gleichung mit z multiplizieren, erhalten wir $1 \equiv x^{\varphi(m)} \pmod{m}$. □