Elementare Zahlentheorie mit Maple (01202) WS 2012/13

Klausur am 16.02.2013:

Musterlösungen

Aufgabe 1

Es gilt $d \mid m$ und $d \mid (m+n)$, also gibt es $x, y \in \mathbb{Z}$ mit $xd = m$ und $yd = m+n$. Wir formen die zweite Gleichung um und erhalten $n = yd - m = yd - xd = d(y-x)$. Es folgt, dass d ein Teiler von n ist.

Aufgabe 2

Da $p_i \neq 2$ für alle $1 \leq i \leq n$ gilt, ist $p_1 p_2 \cdots p_n$ ungerade und damit $p_1 p_2 \cdots p_n + 1$ gerade. $p_1 p_2 \cdots p_n + 1$ ist aber größer als 2 und damit keine Primzahl.

Aufgabe 3

Aus $a^2 \equiv b^2 \pmod{p}$ folgt $p \mid a^2 - b^2 = (a-b)(a+b)$. Da p eine Primzahl ist, folgt $p \mid a-b$, also $a \equiv b \pmod{p}$, oder $p \mid a+b$, also $a \equiv -b \pmod{p}$.

Aufgabe 4

Sei zunächst $n = 1$. Dann folgt $d = 1$ und daraus die Behauptung.

Es seien nun $d = \prod_{i=1}^{k} p_i^{e_i}$ und $n = \prod_{i=1}^{k} p_i^{f_i} \prod_{j=1}^{l} q_j^{h_j}$ die Primfaktorzerlegungen von d und n, wobei die p_i für alle $1 \leq i \leq k$ und die q_j für alle $0 \leq j \leq l$ ($l = 0$ ist zugelassen, falls n eine Primzahlpotenz ist) Primzahlen sind und ferner $1 \leq e_i \leq f_i, h_j \geq 1$ und $q_j \neq p_i$ für alle i, j gilt. Dann folgt

$$\varphi(d) = \prod_{i=1}^{k} p_i^{e_i - 1}(p_i - 1),$$

und

$$\varphi(n) = \varphi(\prod_{i=1}^{k} p_i^{f_i})\varphi(\prod_{j=1}^{l} q_j^{h_j}) = \prod_{i=1}^{k} p_i^{f_i - 1}(p_i - 1) \prod_{j=1}^{l} q_j^{h_j - 1}(q_j - 1).$$

Dann gilt wieder $0 \leq e_i - 1 \leq f_i - 1$, und es folgt $p_i^{e_i - 1}(p_i - 1) \mid p_i^{f_i - 1}(p_i - 1)$ für $1 \leq i \leq k$, also $\varphi(d) \mid \varphi(n)$.

Aufgabe 5

Angenommen, (a, a, c) ist eine Lösung von $X^n + Y^n = Z^n$. Dann gilt $a^n + a^n = 2a^n = c^n$ für natürliche Zahlen a und c. Seien $a = \prod_{i=1}^{r} p_i^{e_i}$ und $c = \prod_{j=1}^{s} q_j^{f_j}$ die Primfaktorzerlegungen von a und c. In den Primfaktorzerlegungen von a^n und c^n sind dann alle Exponenten der Primteiler Vielfache von n. Bei $2a^n$ hingegen ist der Exponent ε von 2 nicht durch n teilbar, es gilt $\varepsilon \equiv 1 \pmod{n}$. Aus der eindeutigen Primfaktorzerlegung von c^n folgt, dass $2a^n \neq c^n$ ist. Also war unsere Annahme falsch.

Aufgabe 6

Wir nehmen an, dass die Gauß'schen Zahlen $1+2i$ und $2+i$ zueinander assoziiert sind. Dann gibt es eine Einheit $e \in \mathbb{Z}[i]$, so dass $1+2i = e(2+i)$ gilt. Da $e \in \{1, -1, i, -i\}$, folgt

$$\begin{aligned} 1 \cdot (2+i) &= 2+i \neq 1+2i \\ (-1) \cdot (2+i) &= -2-i \neq 1+2i \\ i \cdot (2+i) &= -1+2i \neq 1+2i \\ -i \cdot (2+i) &= 1-2i \neq 1+2i \end{aligned}$$

Unsere Annahme war also falsch.

Aufgabe 7

1. Nach Deklaration der lokalen Variablen `k`, `i` und `p` werden diese zunächst mit 1, 2 bzw. 360 initialisiert. Im Anschluss treten wir in eine `while`-Schleife ein, die als Bedingung prüft, ob `p` > 1 ist. Wegen `p` $= 360 > 1$ ist die Bedingung erfüllt und der erste Schleifendurchlauf wird ausgeführt. Die lokale Variable `a` wird mit 0 initialisiert. Es folgt eine weitere `while`-Schleife, die prüft, ob `p` durch `i` $= 2$ teilbar ist. Wegen $360 = 2 \cdot 180$ wird die zweite `while`-Schleife ausgeführt. Es wird `a` $= 1$ und `p` $= 180$ gesetzt. Wir springen an den Anfang der zweiten `while`-Schleife, und wegen $180 = 2 \cdot 90$ wird sie erneut ausgeführt. Es werden `a` $= 2$ und `p` $= 90$ gesetzt. Wieder zurück an den Anfang der zweiten `while`-Schleife, und wegen $90 = 2 \cdot 45$ werden `a` $= 3$ und `p` $= 45$ gesetzt. Zurück am Anfang der zweiten `while`-Schleife sehen wir, dass 45 nicht durch 2 teilbar ist. Also wird die Schleife nicht mehr ausgeführt und wir springen zum `if`-Zweig. `a` $= 3$ ist ungerade, deshalb wird nicht der `then`-Zweig ausgeführt, sondern der `else`-Zweig. Es wird `a` $= \frac{3-1}{2} = 1$ und nach dem `if`-Zweig werden `k` $= 2$ und `i` auf die nächste Primzahl 3 gesetzt. Damit springen wir an den Anfang der ersten `while`-Schleife. Die Prüfung der Bedingung $45 > 1$ ergibt, dass die Schleife erneut ausgeführt wird. Es folgt `a` $= 0$ und wegen $45 = 3 \cdot 15$ erhalten wir `a` $= 1$ und `p` $= 15$. Zurück zum Anfang der zweiten `while`-Schleife sehen wir, dass $15 = 3 \cdot 5$ gilt, also wird die Schleife ausgeführt und werden `a` $= 2$ und `p` $= 5$ gesetzt. Nun ist die Bedingung $5 \bmod 3 = 0$ nicht erfüllt und die zweite `while`-Schleife wird nicht mehr ausgeführt. Mit dem `if`-Zweig erhalten wir `a` $= 1$, denn 2 ist gerade. Nach dem `if`-Zweig werden `k` $= 2 \cdot 3 = 6$ und `i` $= 5$ gesetzt. Da $5 > 1$ gilt, wird die erste `while`-Schleife erneut ausgeführt. Es wird `a` wieder auf 0 gesetzt und wegen $5 = 1 \cdot 5$ folgt `a` $= 1$ und `p` $= 1$. Die zweite `while`-Schleife wird nicht noch einmal ausgeführt und mit dem `if`-Zweig erhalten wir `a` $= 0$. Nach dem `if`-Zweig erhalten wir `k` $= 6$ und `i` wird auf 7 gesetzt. Da nun aber `p` $= 1$ gilt, wird die erste `while`-Schleife nicht mehr ausgeführt. Zum Schluss wird noch `k` $= 6$ ausgegeben.

2. Die Prozedur berechnet das $k \in \mathbb{N}$ aus Lemma 6.2.18, so dass für eine natürliche Zahl n gilt:
$$n = k^2 p_1 \cdots p_r,$$
wobei die $p_1, \ldots, p_r$ verschiedene Primzahlen sind.

Aufgabe 8

Wir deklarieren zuerst die lokalen Variablen `M, i` und `a`, und initialisieren `M` als leere Menge. Mit Hilfe einer `for`-Schleife berechnen wir nun alle Zahlen der Form $n^3 - 1$ für alle $n \in \{1, \ldots, 100\}$ und überprüfen jeweils, ob es sich dabei um eine Primzahl handelt (`if`-Bedingung). Ist dies der Fall, so wird die entsprechende Zahl der Menge `M` hinzugefügt. Nach Beendigung der `for`-Schleife wird die Menge `M` ausgegeben.

Eine mögliche Prozedur könnte folgendermaßen aussehen:

```
> primzahl:=proc() # berechnet alle Primzahlen der Form n^3-1
                   # für n von 1 bis 100, speichert sie in
                   # einer Menge und gibt diese aus
  local M, i, a;
  M:= {};
  for i from 1 to 100 do
    a:=i^3 - 1;
    if isprime(a) then
      M:=M union{a};
    fi;
  od;
  print(M);
  end:
```