MATH 109 – HOMEWORK 5

Due Friday, February 16th. Handwritten submissions only.
The exercises in this homework are worth 16 points.

Exercise 1
Let $a, b \in \mathbb{N}_0$ with $b \neq 0$. Prove that

$$\gcd(a, b) = \gcd\left(b, a - \left\lfloor \frac{a}{b} \right\rfloor b\right).$$

Here, $\lfloor q \rfloor$ denotes the largest integer not larger than $q \in \mathbb{R}$.
Hint: Use a similarly looking result seen previously in the lecture.

Solution 1
From the lecture we recall the following result: for all $a_0, b_0 \in \mathbb{N}_0$, where $a_0 \neq 0$ and $b_0 \neq 0$, for all $q \in \mathbb{N}_0$ we have

$$\gcd(a_0, b_0) = \gcd(b_0, a_0 + q b_0). \tag{1}$$

Now consider the situation in the problem statement: Let $a, b \in \mathbb{N}_0$ with $b \neq 0$. We want to show

$$\gcd(a, b) = \gcd\left(b, a - \left\lfloor \frac{a}{b} \right\rfloor b\right).$$

We define

$$b_0 := b, \quad a_0 := a - \left\lfloor \frac{a}{b} \right\rfloor b.$$

Since $b \in \mathbb{N}$ with $b \neq 0$ by assumption we have that $b_0 \in \mathbb{N}$ with $b_0 \neq 0$ by assumption. Furthermore, we obviously have $a_0 \in \mathbb{Z}$. But also $a_0 \in \mathbb{N}_0$ because

$$a - \left\lfloor \frac{a}{b} \right\rfloor b > a - \frac{a}{b} b = 0.$$

With these choices of a_0 and b_0 and $q := \left\lfloor \frac{a}{b} \right\rfloor$ we then apply the lemma from the lecture, so (1) holds. But we also observe that

$$\gcd(a_0, b_0) = \gcd\left(a - \left\lfloor \frac{a}{b} \right\rfloor b, b\right) = \gcd\left(b, a - \left\lfloor \frac{a}{b} \right\rfloor b\right),$$
$$\gcd(b_0, a_0 + q b_0) = \gcd\left(b, a - \left\lfloor \frac{a}{b} \right\rfloor b + \left\lfloor \frac{a}{b} \right\rfloor b\right) = \gcd(b, a) = \gcd(a, b).$$

This completes the proof.

Exercise 2
Let $a, b \in \mathbb{N}$ such that $a > b$ and let $q, r \in \mathbb{N}_0$ with

$$a = q \cdot b + r, \quad 0 \leq r < b.$$

Show that $r < a/2$.

Solution 2
If $b > a/2$, then $q \in \{0, 1\}$. If namely $q \geq 2$ would hold, then

$$q \cdot b + r \geq q \cdot b \geq 2b > a.$$

Furthermore, since $b < a$, we have $q \geq 1$. In combination, this gives $q = 1$.

Now, if $b > a/2$ we have $q = 1$, and thus $r = a - b$. Now

$$2r = 2a - 2b = 2(a - b) < a,$$

which shows $r < a/2$. So the claim holds in the case.

If instead $b = a/2$, then we have $q = 2$ and thus $r = 0$, and the claim is obviously true too in that case.

Finally, if $b < a/2$, assume that $r \geq a/2$. We then have

$$a = q \cdot b + r = (q + 1) \cdot b + r - b.$$

Since $b < a/2 \leq r$, we get $b > r > r - b > 0$. But since contradicts the uniqueness of the representation of the quotient-remainder theorem. Hence in the case $b < a/2$ we must have $r < a/2$.

This completes the proof.

Exercise 3

Prove the following statement on the run-time of the Euclidean algorithm: if $a, b \in \mathbb{N}$ and $k \in \mathbb{N}_0$ such that $a < 2^k$ and $b < 2^k$, then the Euclidean algorithm takes at most $2k$ steps.

Remark: This statement shows that the Euclidean algorithm has a run-time that growth at most logarithmically in the number of digits of the two input variables.

Solution 3

We prove the statement by induction over k. If $k = 1$, then the statement is true as can be shown by considering the possible inputs.

Let us suppose that the statement is true for some natural number k. We then show that the Euclidean algorithm for all $a, b \in \mathbb{N}$ with $a, b < 2^{k+1}$ takes at most $2(k + 1)$ steps.

The Euclidean algorithm on input a and b either terminates after at two steps or within two steps computes first

$$a_1 := b, \quad b_1 := a - \left\lfloor \frac{a}{b} \right\rfloor b,$$

and then

$$a_2 := b_1, \quad b_2 := a_1 - \left\lfloor \frac{a_1}{b_1} \right\rfloor b_1.$$

It then proceeds by calling itself with input a_2 and b_2. By construction, $a_2 = b_1$ is the remainder of dividing a by b, and thus $a_2 < a/2$. Similarly, b_2 is the remainder of dividing b by b_1, and thus $b_2 < b/2$.

The Euclidean algorithm now proceeds by calling itself with input a_2 and b_2, both of which are not larger than 2^k. By the induction assumption, the Euclidean now requires at most $2k$ steps. In combination with the previous steps, we see that the Euclidean algorithm with input a and b requires at most $2k+2 = 2\,(k + 1)$ steps, and thus the prove is complete.

Exercise 4

Prove the correctness of the naive method to find the greatest common divisor of two numbers. You may restrict yourself to the special case that $a \neq 0$ and $b \neq 0$.

Solution 4

Let $a, b \in \mathbb{N}$. The naive method works as follows:

- Introduce the variable g and set it to 1.
- For $d = 1, \ldots \min(a, b)$ do:
  If $d \mid a$ and $d \mid b$ and $d > g$, then set g equal to d.
- Output g

We first observe that this algorithm terminates after finitely many steps, i.e., your computer doesn't freeze, since the loop in the middle runs only over the finite sequence of numbers $1, \ldots, \min(a, b)$.

Second, we show that the output is always a common divisor of a and b. Indeed, the variable g has initially the value 1, which is clearly a divisor of both a and b. Whenever g gets overwritten with a new value d, then this happens only if $d \mid a$ and $d \mid b$, i.e., g is a divisor of a and b. Hence throughout the run-time of the program, the variable g always contains a divisor of both a and b.

Finally, we show that the algorithm outputs not only some divisor but the greatest common divisor in particular. We write $D := \gcd(a, b)$. Then clearly $1 \leq D \leq \min(a, b)$, and thus throughout the run-time of the program the variable d will at some point in time assume the value D. In that case the program observes that $d \mid a$ and $d \mid b$ and also $d > g$, since then the variable d is the greatest common divisor and g is a different divisor; hence when d holds the value D, the variable g will be overwritten with the value D. So throughout the execution of the program the variable d assumes the value D at some time. Furthermore, it keeps that value because whenever d is any divisor of a and b, the program will recognize that $\neg(d > g)$, so the value of g, already containing the greatest common divisor, will be not be overwritten.

Consequently, the output is the greatest common divisor.

Exercise 5

Prove the following statement: if $A \subseteq \mathbb{N}_0$ and $c \in \mathbb{N}_0$ such that $a \leq c$ for all $a \in A$, then there exists $m \in A$ such that $a \leq m$ for all $a \in A$.

Solution 5

Suppose that $A \subseteq \mathbb{N}_0$ and $c \in \mathbb{N}_0$ such that $a \leq c$ for all $a \in A$, then we define the set

$$B := \{b \in \mathbb{Z} \mid b = c - a\}.$$

Since $a \leq c$ for all $a \in A$ we have that $c - a \in \mathbb{N}$ for all $a \in A$. Thus we see that $B \subseteq \mathbb{N}_0$.

Using the well-ordering principle of the natural numbers, we get that B contains a least element $b_0 \in B$, i.e., we have $b_0 \leq b$ for all $b \in B$.

We define $a_0 := c - b_0$. Obviously, $b_0 = c - a_0$. We show that a_0 is maximal among the members of A. Indeed, if we assume to the contrary that $a \leq A$ with $a_0 < a$, then we have $b := c - a \in B$ and

$$b = c - a < c - a_0 = b_0.$$

But that contradicts b_0 being the minimal element of B. Hence a_0 must be maximal.

Exercise 6

We define a recursive series $a_1, a_2, a_3 \ldots$ as follows. We let $a_1 = a$ for some $a \in \mathbb{N}$ and for all $k \in \mathbb{N}$ we define

$$a_{k+1} = \begin{cases} a_k/2 & \text{if } a_k \text{ is even,} \\ a_k + 1 & \text{if } a_k \text{ is odd.} \end{cases}$$

Show that for every choice of a there exists $k \in \mathbb{N}$ such that $a_k = 1$.

Hint: to get an idea of what is going on, pick some $a \in \mathbb{N}$ and write down a_1, a_2, a_3. For example, try $a = 11, 56, 1025$.

Solution 6

We prove the claim by a variant of induction.

We first observe that the claim is true for $a \in \{1, 2\}$. If $a_1 = 1$, then there is nothing to show, and if $a_1 = 2$, then $a_2 = 1$ by construction.

Next we show if the claim is true for all numbers $a \leq 2^n$ for some fixed $n \in \mathbb{N}$, then the claim is also true for all numbers $a \leq 2^{n+1}$. Let $a \in \mathbb{N}$ with $a \leq 2^{n+1}$. We have $a_1 = a$. If a is even, then $a_2 = a_1/2 \leq 2^{n+1}/2 = 2^n$, and if a is odd, then $a_2 = a_1 + 1$ is even at most 2^{n+1}; thus $a_3 = a_2/2 \leq 2^{n+1}/2 = 2^n$ in that case.

This shows that $a_2 \leq 2^n$ or $a_3 \leq 2^n$. The subsequent entries in the sequence will correspond to the entries of the sequence that would have started with a_2 or a_3, respectively. By the induction hypothesis, there exists $n \in \mathbb{N}$ such that $a_{n+2} = 1$ or $a_{n+3} = 1$. But this just shows that the claim is true for all $a \leq 2^{n+1}$.

We thus have the seen the claim is true for all $a \leq 2$ and that the claim being true for all $a \leq 2^n$ for some $n \in \mathbb{N}$ implies the claim being true for all $a \leq 2^{n+1}$. By the principle of induction, for all $n \in \mathbb{N}$ we have that the claim is true for all $a \leq 2^n$. But since for every $a \in \mathbb{N}$ there exists $n \in \mathbb{N}$ we get that the claim is true for all $a \in \mathbb{N}$.

Exercise 7

We define a recursive series $a_1, a_2, a_3 \ldots$ as follows. We let $a_1 = a$ for some $a \in \mathbb{N}$ and for all $k \in \mathbb{N}$ we define

$$a_{k+1} = \begin{cases} a_k/2 & \text{if } a_k \text{ is even,} \\ 3a_k + 1 & \text{if } a_k \text{ is odd.} \end{cases}$$

Show that for every choice of a there exists $k \in \mathbb{N}$ such that $a_k = 1$. *Hint: this is the Collatz conjecture and generally considered as one of the hardest open problems in mathematics. Don't try to solve it! If you happen to solve it, you will pass this course with A+ and receive a Fields medal.*