Discrete Mathematics in Computer Science

Divisibility

Malte Helmert, Gabriele Röger

University of Basel

Divisibility

- Can we equally share n muffins among m persons without cutting a muffin?

Divisibility

- Can we equally share n muffins among m persons without cutting a muffin?
- If yes then n is a multiple of m and m divides n.

Divisibility

- Can we equally share n muffins among m persons without cutting a muffin?
- If yes then n is a multiple of m and m divides n.
- We consider a generalization of this concept to the integers.

Divisibility

Definition (divisor, multiple)

Let $m, n \in \mathbb{Z}$. If there exists a $k \in \mathbb{Z}$ such that $mk = n$, we say that m divides n, m is a divisor of n or n is a multiple of m and write this as $m \mid n$.

Divisibility

Definition (divisor, multiple)

Let $m, n \in \mathbb{Z}$. If there exists a $k \in \mathbb{Z}$ such that $mk = n$, we say that m divides n, m is a divisor of n or n is a multiple of m and write this as $m \mid n$.

Which of the following are true?

- $2 \mid 4$
- $-2 \mid 4$
- $2 \mid -4$
- $4 \mid 2$
- $3 \mid 4$

Divisibility and Linear Combinations

Theorem (Linear combinations)

Let a, b and d be integers. If $d \mid a$ and $d \mid b$ then for all integers x and y it holds that $d \mid xa + yb$.

Divisibility and Linear Combinations

Theorem (Linear combinations)

Let a, b and d be integers. If $d \mid a$ and $d \mid b$ then for all integers x and y it holds that $d \mid xa + yb$.

Proof.

If $d \mid a$ and $d \mid b$ then there are $k, k' \in \mathbb{Z}$ such that $kd = a$ and $k'd = b$.
It holds that $xa + yb = xkd + yk'd = (xk + yk')d$.
As x, y, k, k' are integers, $xk + yk'$ is integer, thus $d \mid xa + yb$. □

Divisibility and Linear Combinations

Theorem (Linear combinations)

Let a, b and d be integers. If $d \mid a$ and $d \mid b$ then for all integers x and y it holds that $d \mid xa + yb$.

Proof.

If $d \mid a$ and $d \mid b$ then there are $k, k' \in \mathbb{Z}$ such that $kd = a$ and $k'd = b$.
It holds that $xa + yb = xkd + yk'd = (xk + yk')d$.
As x, y, k, k' are integers, $xk + yk'$ is integer, thus $d \mid xa + yb$. □

Some consequences:

- $d \mid a - b$ iff $d \mid b - a$
- If $d \mid a$ and $d \mid b$ then $d \mid a + b$ and $d \mid a - b$.
- If $d \mid a$ then $d \mid -8a$.

Multiplication and Exponentiation

Theorem

Let $a, b, c \in \mathbb{Z}$ *and* $n \in \mathbb{N}_{>0}$.
If $a \mid b$ *then* $ac \mid bc$ *and* $a^n \mid b^n$.

Multiplication and Exponentiation

Theorem

Let $a, b, c \in \mathbb{Z}$ *and* $n \in \mathbb{N}_{>0}$*.*
If $a \mid b$ *then* $ac \mid bc$ *and* $a^n \mid b^n$*.*

Proof.

If $a \mid b$ there is a $k \in \mathbb{Z}$ such that $ak = b$.

Multiplication and Exponentiation

Theorem

Let $a, b, c \in \mathbb{Z}$ and $n \in \mathbb{N}_{>0}$.
If $a \mid b$ then $ac \mid bc$ and $a^n \mid b^n$.

Proof.

If $a \mid b$ there is a $k \in \mathbb{Z}$ such that $ak = b$.

Multiplying both sides with c, we get $cak = cb$ and thus $ca \mid cb$.

Multiplication and Exponentiation

Theorem

Let $a, b, c \in \mathbb{Z}$ *and* $n \in \mathbb{N}_{>0}$.
If $a \mid b$ *then* $ac \mid bc$ *and* $a^n \mid b^n$.

Proof.

If $a \mid b$ there is a $k \in \mathbb{Z}$ such that $ak = b$.

Multiplying both sides with c, we get $cak = cb$ and thus $ca \mid cb$.

From $ak = b$, we also get $b^n = (ak)^n = a^n k^n$, so $a^n \mid b^n$. □

Partial Order

If we consider only the natural numbers, divisibility is a partial order:

Theorem

Divisibility $\mid$ over $\mathbb{N}_0$ is a partial order.

Proof.

...

Partial Order

If we consider only the natural numbers,
divisibility is a partial order:

Theorem

Divisibility $|$ over $\mathbb{N}_0$ is a partial order.

Proof.

- reflexivity: For all $m \in \mathbb{N}_0$ it holds that $m \cdot 1 = m$, so $m \mid m$.

...

Partial Order

If we consider only the natural numbers, divisibility is a partial order:

Theorem

Divisibility $\mid$ over $\mathbb{N}_0$ is a partial order.

Proof.

- reflexivity: For all $m \in \mathbb{N}_0$ it holds that $m \cdot 1 = m$, so $m \mid m$.
- transitivity: If $m \mid n$ and $n \mid o$ there are $k, k' \in \mathbb{Z}$ such that $mk = n$ and $nk' = o$. With $k'' = kk'$ it holds then that $o = nk' = mkk' = mk''$, and consequently $m \mid o$.

...

Partial Order

Proof (continued).

- antisymmetry: We show that if $m \mid n$ and $n \mid m$ then $m = n$.

Partial Order

Proof (continued).

- antisymmetry: We show that if $m \mid n$ and $n \mid m$ then $m = n$. If $m = n = 0$, there is nothing to show.

Partial Order

Proof (continued).

- antisymmetry: We show that if $m \mid n$ and $n \mid m$ then $m = n$.

  If $m = n = 0$, there is nothing to show.

  Otherwise, at least one of m and n is positive.

Partial Order

Proof (continued).

- antisymmetry: We show that if $m \mid n$ and $n \mid m$ then $m = n$.

  If $m = n = 0$, there is nothing to show.

  Otherwise, at least one of m and n is positive.

  Let this w.l.o.g. (without loss of generality) be m.

Partial Order

Proof (continued).

- antisymmetry: We show that if $m \mid n$ and $n \mid m$ then $m = n$.

  If $m = n = 0$, there is nothing to show.

  Otherwise, at least one of m and n is positive.

  Let this w.l.o.g. (without loss of generality) be m.
  If $m \mid n$ and $n \mid m$ then there are $k, k' \in \mathbb{Z}$
  such that $mk = n$ and $nk' = m$.

Partial Order

Proof (continued).

- antisymmetry: We show that if $m \mid n$ and $n \mid m$ then $m = n$.

  If $m = n = 0$, there is nothing to show.

  Otherwise, at least one of m and n is positive.

  Let this w.l.o.g. (without loss of generality) be m.
  If $m \mid n$ and $n \mid m$ then there are $k, k' \in \mathbb{Z}$
  such that $mk = n$ and $nk' = m$.

  Combining these, we get $m = nk' = mkk'$, which implies
  (with $m \neq 0$) that $kk' = 1$.

Partial Order

Proof (continued).

- antisymmetry: We show that if $m \mid n$ and $n \mid m$ then $m = n$.

  If $m = n = 0$, there is nothing to show.

  Otherwise, at least one of m and n is positive.

  Let this w.l.o.g. (without loss of generality) be m.
  If $m \mid n$ and $n \mid m$ then there are $k, k' \in \mathbb{Z}$ such that $mk = n$ and $nk' = m$.

  Combining these, we get $m = nk' = mkk'$, which implies (with $m \neq 0$) that $kk' = 1$.

  Since k and k' are integers, this implies $k = k' = 1$ or $k = k' = -1$. As $mk = n$, m is positive and n is non-negative, we can conclude that $k = 1$ and $m = n$.

□

Discrete Mathematics in Computer Science

Modular Arithmetic

Malte Helmert, Gabriele Röger

University of Basel

Halloween is Coming

- You have m sweets.
- There are k kids showing up for trick-or-treating.
- To keep everything fair, every kid gets the same amount of treats.
- You may enjoy the rest. :-)
- How much does every kid get, how much do you get?

Euclid's Division Lemma

Theorem (Euclid's division lemma)

For all integers a and b with $b \neq 0$
there are unique integers q and r
with $a = qb + r$ and $0 \leq r < |b|$.

Number q is called the quotient and r the remainder.

Without proof.

Euclid's Division Lemma

Theorem (Euclid's division lemma)

For all integers a and b with $b \neq 0$
there are unique integers q and r
with $a = qb + r$ and $0 \leq r < |b|$.

Number q is called the quotient and r the remainder.

Without proof.

Examples:

- $a = 18, b = 5$
- $a = 5, b = 18$
- $a = -18, b = 5$
- $a = 18, b = -5$

Modulo Operation

- With $a \bmod b$ we refer to the remainder of Euclidean division.

Modulo Operation

- With *a* mod *b* we refer to the remainder of Euclidean division.
- Most programming languages have a built-in operator to compute *a* mod *b* (for positive integers):

```
int mod = 34 % 7;
// result 6 because 4 * 7 + 6 = 34
```

Modulo Operation

- With $a \bmod b$ we refer to the remainder of Euclidean division.
- Most programming languages have a built-in operator to compute $a \bmod b$ (for positive integers):

```
int mod = 34 % 7;
// result 6 because 4 * 7 + 6 = 34
```

- Common application: Determine whether a natural number n is even.

```
n % 2 == 0
```

Modulo Operation

- With *a* mod *b* we refer to the remainder of Euclidean division.
- Most programming languages have a built-in operator to compute *a* mod *b* (for positive integers):

```
int mod = 34 % 7;
// result 6 because 4 * 7 + 6 = 34
```

- Common application: Determine whether a natural number *n* is even.

```
n % 2 == 0
```

- Languages behave differently with negative operands!

Halloween

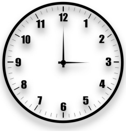
```
def share_sweets(no_kids, no_sweets):
    print("Each kid gets",
          no_sweets // no_kids,
          "of the sweets.")
    print("You may keep",
          no_sweets % no_kids,
          "of the sweets.")
```

Congruence Modulo n

- We now are no longer interested in the value of the remainder but will consider numbers a and a' as equivalent if the remainder with division by a given number b is equal.

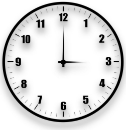
Congruence Modulo n

- We now are no longer interested in the value of the remainder but will consider numbers a and a' as equivalent
  if the remainder with division by a given number b is equal.
- Consider the clock:



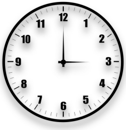
Congruence Modulo n

- We now are no longer interested in the value of the remainder but will consider numbers a and a' as equivalent if the remainder with division by a given number b is equal.
- Consider the clock:
  - It's now 3 o'clock



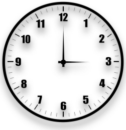
Congruence Modulo n

- We now are no longer interested in the value of the remainder but will consider numbers a and a' as equivalent if the remainder with division by a given number b is equal.
- Consider the clock:
  - It's now 3 o'clock
  - In 12 hours its 3 o'clock



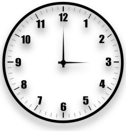
Congruence Modulo n

- We now are no longer interested in the value of the remainder but will consider numbers a and a' as equivalent if the remainder with division by a given number b is equal.
- Consider the clock:
  - It's now 3 o'clock
  - In 12 hours its 3 o'clock
  - Same in 24, 36, 48, ... hours.



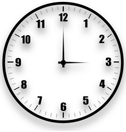
Congruence Modulo n

- We now are no longer interested in the value of the remainder but will consider numbers a and a' as equivalent if the remainder with division by a given number b is equal.
- Consider the clock:
  - It's now 3 o'clock
  - In 12 hours its 3 o'clock
  - Same in 24, 36, 48, ... hours.
  - 15:00 and 3:00 are shown the same.



Congruence Modulo n

- We now are no longer interested in the value of the remainder but will consider numbers a and a' as equivalent
  if the remainder with division by a given number b is equal.
- Consider the clock:
  - It's now 3 o'clock
  - In 12 hours its 3 o'clock
  - Same in 24, 36, 48, . . . hours.
  - 15:00 and 3:00 are shown the same.
  - In the following, we will express this as $3 \equiv 15 \pmod{12}$



Congruence Modulo n – Definition

Definition (Congruence modulo n)

For integer $n > 1$, two integers a and b
are called congruent modulo n if $n \mid a - b$.

We write this as $a \equiv b \pmod{n}$.

Congruence Modulo n – Definition

Definition (Congruence modulo n)

For integer $n > 1$, two integers a and b are called congruent modulo n if $n \mid a - b$.

We write this as $a \equiv b \pmod{n}$.

Which of the following statements are true?

- $0 \equiv 5 \pmod{5}$
- $1 \equiv 6 \pmod{5}$
- $4 \equiv 14 \pmod{5}$
- $-8 \equiv 7 \pmod{5}$
- $2 \equiv -3 \pmod{5}$

Congruence Modulo n – Definition

Definition (Congruence modulo n)

For integer $n > 1$, two integers a and b are called congruent modulo n if $n \mid a - b$.

We write this as $a \equiv b \pmod{n}$.

Which of the following statements are true?

- $0 \equiv 5 \pmod{5}$
- $1 \equiv 6 \pmod{5}$
- $4 \equiv 14 \pmod{5}$
- $-8 \equiv 7 \pmod{5}$
- $2 \equiv -3 \pmod{5}$

Why is this the same concept as described in the clock example?!?

Congruence Corresponds to Equal Remainders

Theorem

For integers a and b and integer $n > 1$ it holds that
$a \equiv b \pmod{n}$ *iff there are* $q, q', r \in \mathbb{Z}$ *with*

$$a = qn + r$$
$$b = q'n + r.$$

Congruence Corresponds to Equal Remainders

Theorem

For integers a and b and integer $n > 1$ it holds that $a \equiv b \pmod{n}$ iff there are $q, q', r \in \mathbb{Z}$ with

$$a = qn + r$$
$$b = q'n + r.$$

Proof sketch.

"$\Rightarrow$": If $n \mid a - b$ then there is a $k \in \mathbb{Z}$ with $kn = a - b$.

Congruence Corresponds to Equal Remainders

Theorem

For integers a and b and integer $n > 1$ it holds that
$a \equiv b \pmod{n}$ *iff there are* $q, q', r \in \mathbb{Z}$ *with*

$$a = qn + r$$
$$b = q'n + r.$$

Proof sketch.

"$\Rightarrow$": If $n \mid a - b$ then there is a $k \in \mathbb{Z}$ with $kn = a - b$.

As $n \neq 0$, by Euclid's lemma there are $q, q', r, r' \in \mathbb{Z}$ with
$a = qn + r$ and $b = q'n + r'$, where $0 \leq r < |n|$ and $0 \leq r' < |n|$.

Congruence Corresponds to Equal Remainders

Theorem

For integers a and b and integer $n > 1$ it holds that $a \equiv b \pmod{n}$ iff there are $q, q', r \in \mathbb{Z}$ with

$$a = qn + r$$
$$b = q'n + r.$$

Proof sketch.

"$\Rightarrow$": If $n \mid a - b$ then there is a $k \in \mathbb{Z}$ with $kn = a - b$.

As $n \neq 0$, by Euclid's lemma there are $q, q', r, r' \in \mathbb{Z}$ with $a = qn + r$ and $b = q'n + r'$, where $0 \leq r < |n|$ and $0 \leq r' < |n|$.

Together, we get that $kn = qn + r - (q'n + r')$, which is the case iff $kn + r' = (q - q')n + r$. By Euclid's lemma, quotients and remainders are unique, so in particular $r' = r$.

Congruence Corresponds to Equal Remainders

Theorem

For integers a and b and integer $n > 1$ it holds that $a \equiv b \pmod{n}$ iff there are $q, q', r \in \mathbb{Z}$ with

$$a = qn + r$$
$$b = q'n + r.$$

Proof sketch.

"$\Rightarrow$": If $n \mid a - b$ then there is a $k \in \mathbb{Z}$ with $kn = a - b$.

As $n \neq 0$, by Euclid's lemma there are $q, q', r, r' \in \mathbb{Z}$ with $a = qn + r$ and $b = q'n + r'$, where $0 \leq r < |n|$ and $0 \leq r' < |n|$.

Together, we get that $kn = qn + r - (q'n + r')$, which is the case iff $kn + r' = (q - q')n + r$. By Euclid's lemma, quotients and remainders are unique, so in particular $r' = r$.

"$\Leftarrow$": If we subtract the equations, we get $a - b = (q - q')n$, so $n \mid a - b$ and $a \equiv b \pmod{n}$.

Congruence Modulo n is an Equivalence Relation

Theorem

Congruence modulo n is an equivalence relation.

Proof sketch.

Congruence Modulo n is an Equivalence Relation

Theorem

Congruence modulo n is an equivalence relation.

Proof sketch.

Reflexive: $a \equiv a \pmod{n}$ because every integer divides 0.

Congruence Modulo n is an Equivalence Relation

Theorem

Congruence modulo n is an equivalence relation.

Proof sketch.

Reflexive: $a \equiv a \pmod{n}$ because every integer divides 0.

Symmetric: $a \equiv b \pmod{n}$ iff $n \mid a - b$ iff $n \mid b - a$ iff $b \equiv a \pmod{n}$.

Congruence Modulo n is an Equivalence Relation

Theorem

Congruence modulo n is an equivalence relation.

Proof sketch.

Reflexive: $a \equiv a \pmod{n}$ because every integer divides 0.

Symmetric: $a \equiv b \pmod{n}$ iff $n \mid a - b$ iff $n \mid b - a$ iff $b \equiv a \pmod{n}$.

Transitive: If $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ then $n \mid a - b$ and $n \mid b - c$. Together, these imply that $n \mid a - b + b - c$. From $n \mid a - c$ we get $a \equiv c \pmod{n}$.

Congruence Modulo n is an Equivalence Relation

Theorem

Congruence modulo n is an equivalence relation.

Proof sketch.

Reflexive: $a \equiv a \pmod{n}$ because every integer divides 0.

Symmetric: $a \equiv b \pmod{n}$ iff $n \mid a - b$ iff $n \mid b - a$ iff $b \equiv a \pmod{n}$.

Transitive: If $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ then $n \mid a - b$ and $n \mid b - c$. Together, these imply that $n \mid a - b + b - c$. From $n \mid a - c$ we get $a \equiv c \pmod{n}$.

For modulus n, the equivalence class of a is
$\bar{a}_n = \{\ldots, a - 2n, a - n, a, a + n, a + 2n, \ldots\}$.
Set $\bar{a}_n$ is called the congruence class or residue of a modulo n.

Compatibility with Operations

Theorem

Congruence modulo n is compatible with addition, subtraction, multiplication, translation, scaling and exponentiation, i. e. if $a \equiv b \pmod{n}$ *and* $a' \equiv b' \pmod{n}$ *then*

- $a + a' \equiv b + b' \pmod{n}$,
- $a - a' \equiv b - b' \pmod{n}$,
- $aa' \equiv bb' \pmod{n}$,
- $a + k \equiv b + k \pmod{n}$ *for all* $k \in \mathbb{Z}$,
- $ak \equiv bk \pmod{n}$ *for all* $k \in \mathbb{Z}$, *and*
- $a^k \equiv b^k \pmod{n}$ *for all* $k \in \mathbb{N}_0$.

Compatibility with Operations

Theorem

Congruence modulo n is compatible with addition, subtraction, multiplication, translation, scaling and exponentiation, i. e. if $a \equiv b \pmod{n}$ and $a' \equiv b' \pmod{n}$ then

- $a + a' \equiv b + b' \pmod{n}$,
- $a - a' \equiv b - b' \pmod{n}$,
- $aa' \equiv bb' \pmod{n}$,
- $a + k \equiv b + k \pmod{n}$ *for all* $k \in \mathbb{Z}$,
- $ak \equiv bk \pmod{n}$ *for all* $k \in \mathbb{Z}$, *and*
- $a^k \equiv b^k \pmod{n}$ *for all* $k \in \mathbb{N}_0$.

Congruence modulo n is a so-called congruence relation
($=$ equivalence relation compatible with operations).

Fermat's Little Theorem

Theorem (Fermat's Little Theorem)

If $a \in \mathbb{Z}$ is not a multiple of prime number p
then $a^{p-1} \equiv 1 \pmod{p}$.

Without proof.

Fermat's Little Theorem

Theorem (Fermat's Little Theorem)

If $a \in \mathbb{Z}$ is not a multiple of prime number p then $a^{p-1} \equiv 1 \pmod{p}$.

Without proof.

Helps finding the remainder when dividing a very large number by a prime number.

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67,
so we can use the theorem.

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67,
so we can use the theorem.

By the theorem, $4^{66} \equiv 1 \pmod{67}$.

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67,
so we can use the theorem.

By the theorem, $4^{66} \equiv 1 \pmod{67}$. How does this help?

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67,
so we can use the theorem.

By the theorem, $4^{66} \equiv 1 \pmod{67}$. How does this help?

Raise both sides to a higher power.
$100000/66 = 1515.\overline{15} \quad \rightarrow$ use 1515

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67,
so we can use the theorem.

By the theorem, $4^{66} \equiv 1 \pmod{67}$. How does this help?

Raise both sides to a higher power.
$100000/66 = 1515.\overline{15} \quad \rightarrow$ use 1515

$(4^{66})^{1515} \equiv 1^{1515} \pmod{67}$

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67,
so we can use the theorem.

By the theorem, $4^{66} \equiv 1 \pmod{67}$. How does this help?

Raise both sides to a higher power.
$100000/66 = 1515.\overline{15}$ $\quad \rightarrow$ use 1515

$(4^{66})^{1515} \equiv 1^{1515} \pmod{67}$ iff
$4^{99990} \equiv 1 \pmod{67}$

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67,
so we can use the theorem.

By the theorem, $4^{66} \equiv 1 \pmod{67}$. How does this help?

Raise both sides to a higher power.
$100000/66 = 1515.\overline{15} \quad \rightarrow$ use 1515

$(4^{66})^{1515} \equiv 1^{1515} \pmod{67}$ iff
$4^{99990} \equiv 1 \pmod{67}$ iff
$4^{10}4^{99990} \equiv 4^{10} \pmod{67}$

Fermat's Little Theorem – Application

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67,
so we can use the theorem.

By the theorem, $4^{66} \equiv 1 \pmod{67}$. How does this help?

Raise both sides to a higher power.
$100000/66 = 1515.\overline{15} \quad \rightarrow$ use 1515

$(4^{66})^{1515} \equiv 1^{1515} \pmod{67}$ iff
$4^{99990} \equiv 1 \pmod{67}$ iff
$4^{10}4^{99990} \equiv 4^{10} \pmod{67}$ iff (calculator)
$4^{100000} \equiv 26 \pmod{67}$