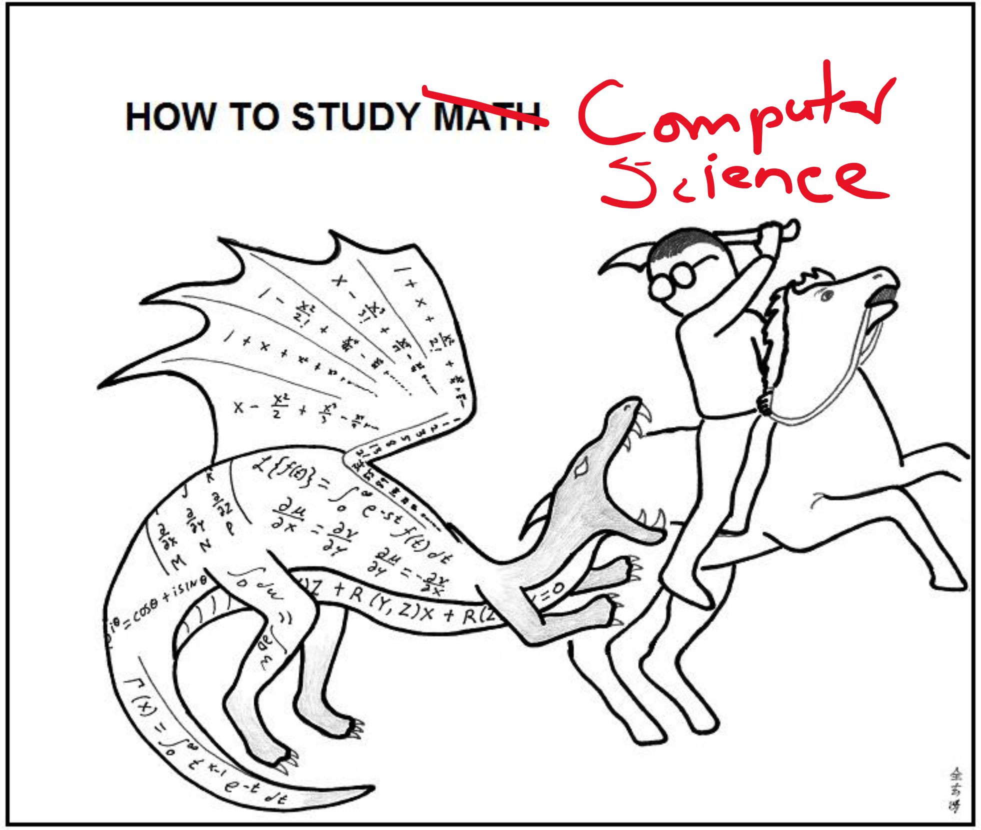


Don't just read it; fight it!

--- Paul R. Halmos

https://abstrusegoose.com/353

Number Theory

CSE 311 Spring 2022
Lecture 11

Proof By Cases

Let $A = \{x : \texttt{Prime}(x)\}$, $B = \{x: \texttt{Odd}(x) \vee \texttt{PowerOfTwo}(x)\}$

Where $\texttt{PowerOfTwo}(x) := \exists c(\texttt{Integer}(c) \wedge x = 2\text{^}c)$

Prove $A \subseteq B$

$\forall x (x \in A \to x \in B)$

We need two different arguments – one for 2 and one for all the other primes…

arb $x \in A$

$\vdots$

$x \in B$

Proof By Cases

Let x be an arbitrary element of A.

We divide into two cases.

Case 1: x is even

If x is even and an element of A (i.e. both even and prime) it must be 2.
So it equals 2^c for $c = 1$, and thus is in B by definition of B.

Case 2: x is odd

Then $x \in B$ by satisfying the first requirement in the definition of B.

In either case, $x \in B$. Since an arbitrary element of A is also in B, we have $A \subseteq B$.

Proof By Cases (Skeleton)

We divide into cases based on []

Case 1: [condition for case 1]

[suppose condition and arrive at target]

Case 2: [condition for case 2]

[suppose condition and arrive at target]

[more cases if necessary]

In every case we have [target].

Often the [target] is an intermediate one and you need a few more steps and/or a conclusion.

Proof By Cases

Make it clear how you decide which case your in.

It should be obvious your cases are “exhaustive”

Reach the same conclusion in each of the cases, and you can say you’ve got that conclusion no matter what (outside the cases).

Advanced version: sometimes you end up arguing a certain case “can’t happen”

Two More Set Operations

Given a set, let's talk about it's powerset.

$\mathcal{P}(A)$

$\mathcal{P}(A) = \{X : X \text{ is a subset of } A\}$

The powerset of A is the **set** of all subsets of A.

$\{1\} \in \mathcal{P}(\{1,2\})$

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

$\{\{\}, \{1\}, \{2\}, \{1,2\}\}$

Two More Set Operations

$A \times B = \{(a, b): a \in A \wedge b \in B\}$

Called "the Cartesian product" of A and B.

$\mathbb{R}^2$

$\mathbb{R} \times \mathbb{R}$ is the "real plane" ordered pairs of real numbers.

$\{1,2\} \times \{1,2,3\} = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3)\}$

$(2,1)$

Read on Your Own

Some old friends (and some new ones)

$\mathbb{N}$ is the set of **Natural Numbers**; $\mathbb{N}$ = {0, 1, 2, …}
$\mathbb{Z}$ is the set of **Integers**; $\mathbb{Z}$ = {…, -2, -1, 0, 1, 2, …}
$\mathbb{Q}$ is the set of **Rational Numbers**; e.g. ½, -17, 32/48
$\mathbb{R}$ is the set of **Real Numbers**; e.g. 1, -17, 32/48, π, $\sqrt{2}$
[n] is the set {1, 2, …, n} when n is a positive integer
{} = ∅ is the **empty set**; the *only* set with no elements

Some old friends (and some new ones)

Our natural numbers start at 0.
Common in CS, other resources start at 1.

$\mathbb{N}$ is the set of **Natural Numbers; $\mathbb{N}$** = {0, 1, 2, …}
$\mathbb{Z}$ is the set of **Integers**; **$\mathbb{Z}$** = {…, -2, -1, 0, 1, 2, …}
$\mathbb{Q}$ is the set of **Rational Numbers**; e.g. ½, -17, 32/48
$\mathbb{R}$ is the set of **Real Numbers**; e.g. 1, -17, 32/48, π, $\sqrt{2}$
[n] is the set **{1, 2, …, n}** when **n** is a positive integer
{} = ∅ is the **empty set**; the *only* set with no elements

In LaTeX \mathbb{R}
In Office \doubleR

Use this symbol not {}.
In LaTex \varnothing In Office \emptyset.

More connectors!

$A \setminus B$ "A minus B"

$$A \setminus B = \{x : x \in A \land x \notin B\}$$

$A \oplus B$ "XOR" (also called "symmetric difference")

$$A \oplus B = \{x : x \in A \oplus x \in B\}$$

Number Theory

Why Number Theory?

Applicable in Computer Science

"hash functions" (you'll see them in 332) commonly use modular arithmetic
Much of classical cryptography is based on prime numbers.

More importantly, a great playground for writing English proofs.

Framing Device

We're going to give you enough background to (mostly) understand the RSA encryption system.

Key generation [edit]

The keys for the RSA algorithm are generated in the following way:

1. Choose two distinct prime numbers p and q.
   - For security purposes, the integers p and q should be chosen at random and should be similar in magnitude but differ in length by a few digits to make factoring harder.[2] Prime integers can be efficiently found using a primality test.
   - p and q are kept secret.
2. Compute $n = pq$.
   - n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
   - n is released as part of the public key.
3. Compute $\lambda(n)$, where λ is Carmichael's totient function. Since $n = pq$, $\lambda(n) = \operatorname{lcm}(\lambda(p), \lambda(q))$, and since p and q are prime, $\lambda(p) = \varphi(p) = p - 1$, and likewise $\lambda(q) = q - 1$. Hence $\lambda(n) = \operatorname{lcm}(p - 1, q - 1)$.
   - $\lambda(n)$ is kept secret.
   - The lcm may be calculated through the Euclidean algorithm, since $\operatorname{lcm}(a, b) = |ab|/\gcd(a, b)$.
4. Choose an integer e such that $1 < e < \lambda(n)$ and $\gcd(e, \lambda(n)) = 1$; that is, e and $\lambda(n)$ are coprime.
   - e having a short bit-length and small Hamming weight results in more efficient encryption – the most commonly chosen value for e is $2^{16} + 1 = 65\,537$. The smallest (and fastest) possible value for e is 3, but such a small value for e has been shown to be less secure in some settings.[15]
   - e is released as part of the public key.
5. Determine d as $d \equiv e^{-1} \pmod{\lambda(n)}$; that is, d is the modular multiplicative inverse of e modulo $\lambda(n)$.
   - This means: solve for d the equation $d \cdot e \equiv 1 \pmod{\lambda(n)}$; d can be computed efficiently by using the extended Euclidean algorithm, since, thanks to e and $\lambda(n)$ being coprime, said equation is a form of Bézout's identity, where d is one of the coefficients.
   - d is kept secret as the *private key exponent*.

The *public key* consists of the modulus n and the public (or encryption) exponent e. The *private key* consists of the private (or decryption) exponent d, which must be kept secret. p, q, and $\lambda(n)$ must also be kept secret because they can be used to calculate d. In fact, they can all be discarded after d has been computed.[16]

Framing Device

We're going to give you enough background to (mostly) understand the RSA encryption system.

Key generation [edit]

Prime Numbers

The keys for the RSA algorithm are genera

1. Choose two distinct prime numbers *p* and *q*.
   - For security purposes, the integers *p* and *q* should be chosen at random and should be similar in magnitude but differ in length by a few digits to make factoring harder.[2] Prime integers can be efficiently found using a primality test.
   - *p* and *q* are kept secret.

Modular Arithmetic

2. Compute $n = pq$.
   - *n* is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
   - *n* is released as part of the public key.
3. Compute $\lambda(n)$, where λ is Carmichael's totient function. Since $n = pq$, $\lambda(n) = \mathrm{lcm}(\lambda(p), \lambda(q))$, and since *p* and *q* are prime, $\lambda(p) = \varphi(p) = p - 1$, and likewise $\lambda(q) = q - 1$. Hence $\lambda(n) = \mathrm{lcm}(p - 1, q - 1)$.
   - $\lambda(n)$ is kept secret.
   - The lcm may be calculated through the Euclidean algorithm, since lcm(*a*, *b*)

Modular Multiplicative Inverse

4. Choose an integer *e* such that $1 < e < \lambda(n)$ and $\gcd(e, \lambda(n)) = 1$; that is, *e* and $\lambda(n)$ are cop
   - *e* having a short bit-length and small Hamming weight results in more efficient enc ne most commonly chosen value for *e* is $2^{16} + 1 = 65\,537$. The smallest (and fastest) possible value for *e* is 3, but such a small value for *e* has been shown to be less secure in some settings.[15]
   - *e* is released as part of the public key.

Bezout's Theorem

5. Determine *d* as $d \equiv e^{-1} \pmod{\lambda(n)}$; that is, *d* is the modular multiplicative inverse of *e* modulo $\lambda(n)$.
   - This means: solve for *d* the equation $d \cdot e \equiv 1 \pmod{\lambda(n)}$; *d* can be computed efficiently by using the extended Euclidean algorithm, since, thanks to *e* and $\lambda(n)$ being coprime, said equation is a form of Bézout's identity, where *d* is one of the coefficients.
   - *d* is kept secret as the *private key exponent*.

Extended Euclidian Algorithm

The *public key* consists of the modulus *n* and the public (or encryption) exponent *e*. The *private key* consists of the private (or decryp also be kept secret because they can be used to calculate *d*. In fact, they can all be discarded after *d* has been computed.[16]

Framing Device

We're going to give you enough background to (mostly) understand the RSA encryption system.

Encryption [edit]

After Bob obtains Alice's public key, he can send a message M to Alice.

To do it, he first turns M (strictly speaking, the un-padded plaintext) into an integer m (strictly speaking, the padded plaintext), such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c, using Alice's public key e, corresponding to

$$c \equiv m^e \pmod{n}.$$

This can be done reasonably quickly, even for very large numbers, using modular exponentiation. Bob then transmits c to Alice. Note that at least nine values of *m* will yield a ciphertext c equal to *m*,[22] but this is very unlikely to occur in practice.

Decryption [edit]

Alice can recover m from c by using her private key exponent d by computing

$$c^d \equiv (m^e)^d \equiv m \pmod{n}.$$

Given m, she can recover the original message M by reversing the padding scheme.

Framing Device

We're going to give you enough background to (mostly) understand the RSA encryption system.

Encryption [edit]

After Bob obtains Alice's public key, he can send a message M to Alice.

To do it, he first turns M (strictly speaking, the un-padded plaintext) into an integer m (strictly speaking, the padded plaintext), such that $0 \le m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c, using Alice's public key e, corresponding to

$$c \equiv m^e \pmod{n}.$$

This can be done reasonably quickly, even for very large numbers, using modular exponentiation. Bob then transmits c to Alice. Note that at least nine values of *m* will yield a ciphertext c equal to *m*,[22] but this is very unlikely to occur in practice.

Modular Exponentiation

Decryption [edit]

Alice can recover m from c by using her private key exponent d by computing

$$c^d \equiv (m^e)^d \equiv m \pmod{n}.$$

Given m, she can recover the original message M by reversing the padding scheme.

Divides

$x|y$ $3|6$

Divides

For integers x, y we say $x|y$ ("x divides y") iff there is an integer z such that $xz = y$.

$xz = y$

$yz = x$

"x is a divisor of y" or "x is a factor of y" means (essentially) the same thing as x divides y.

("essentially" because of edge cases like when a number is negative or $y = 0$)

"The small number goes first"

Divides

Divides

For integers x, y we say $x|y$ ("x divides y") iff there is an integer z such that $xz = y$.

Which of these are true?

$2|4$ True $2 \cdot 2 = 4$

$4|2$ False

$2 \cdot ? = -2$

$2|-2$ True

$5|0$ True

$0|5$ False

$1|5$ True

Divides

Divides

For integers x, y we say $x|y$ ("x divides y") iff there is an integer z such that $xz = y$.

Which of these are true?

$2|4$	True	$4|2$	False	$2|-2$	True
$5|0$	True	$0|5$	False	$1|5$	True

A useful theorem

$\mathbb{Z} = \{\text{integers}\}$

The Division Theorem

For every $a \in \mathbb{Z}$, $\boldsymbol{d} \in \mathbb{Z}$ with $\boldsymbol{d} > \mathbf{0}$

There exist *unique* integers q, r with $0 \le r < d$

Such that $a = dq + r$

Remember when non integers were still secret, you did division like this?

d q 4 R5 r

7)33 a

28

5

q is the “quotient”

r is the “remainder”

$$\frac{a}{d} = q + \frac{r}{d}$$

Unique

The Division Theorem

For every $a \in \mathbb{Z}$, $\boldsymbol{d} \in \mathbb{Z}$ with $\boldsymbol{d} > \boldsymbol{0}$
There exist *unique* integers q, r with $0 \leq r < d$
Such that $a = dq + r$

"unique" means "only one"....but be careful with how this word is used.

r is unique, **given** a, d. – it still depends on a, d but once you've chosen a and d

"unique" is not saying $\exists r \forall a, d \;\; P(a, d, r)$
It's saying $\forall a, d \exists r [P(a, d, r) \wedge [P(a, d, x) \rightarrow x = r]]$

A useful theorem

The Division Theorem

For every $a \in \mathbb{Z}$, $\boldsymbol{d} \in \mathbb{Z}$ with $\boldsymbol{d > 0}$
There exist *unique* integers q, r with $0 \leq r < d$
Such that $a = dq + r$

The q is the result of `a/d` (integer division) in Java

The r is the result of `a%d` in Java

That's slightly a lie, r is always non-negative, Java's % operator sometimes gives a negative number.

Terminology

"remainder"

You might have called the % operator in Java "mod"

We're going to use the word "mod" to mean a closely related, but different thing.

Java's % is an operator (like + or ·) you give it two numbers, it produces a number.

The word "mod" in this class, refers to a set of rules

Modular Arithmetic

“arithmetic mod 12” is familiar to you. You do it with clocks.

What’s 3 hours after 10 o’clock?

1 o’clock. You hit 12 and then “wrapped around”

“13 and 1 are the same, mod 12” “-11 and 1 are the same, mod 12”

We don’t just want to do math for clocks – what about if we need to talk about parity (even vs. odd) or ignore higher-order-bits (mod by 16, for example)

Modular Arithmetic

To say "the same" we don't want to use ~~=~~ … that means the normal =

We'll write $13 \equiv 1 \pmod{12}$

≡ because "equivalent" is "like equal," and the "modulus" we're using in parentheses at the end so we don't forget it.

(we'll also say "congruent mod 12")

The notation here is bad. We all agree it's bad. Most people still use it.

$13 \equiv_{12} 1$ would have been better. "mod 12" is giving you information about the ≡ symbol, it's not operating on 1.

Modular Arithmetic

$a \equiv b \pmod{n}$

We need a definition! We can't just say "it's like a clock"

Pause what do you expect the definition to be?

Is it related to % ?

$a \% n = b \% n$

Modular Arithmetic

We need a definition! We can't just say "it's like a clock"

$a \% n = b \% n$

Pause what do you expect the definition to be?

Equivalence in modular arithmetic

Let $a \in \mathbb{Z}, b \in \mathbb{Z}, n \in \mathbb{Z}$ and $n > 0$.
We say $a \equiv b \ (mod\ n)$ if and only if $n | (b - a)$

Huh?

Long Pause

It’s easy to read something with a bunch of symbols and say “yep, those are symbols.” and keep going

STOP Go Back.

You have to *fight* the symbols they’re probably trying to pull a fast one on you.

Same goes for when I’m presenting a proof – you shouldn’t just believe me – I’m wrong all the time!

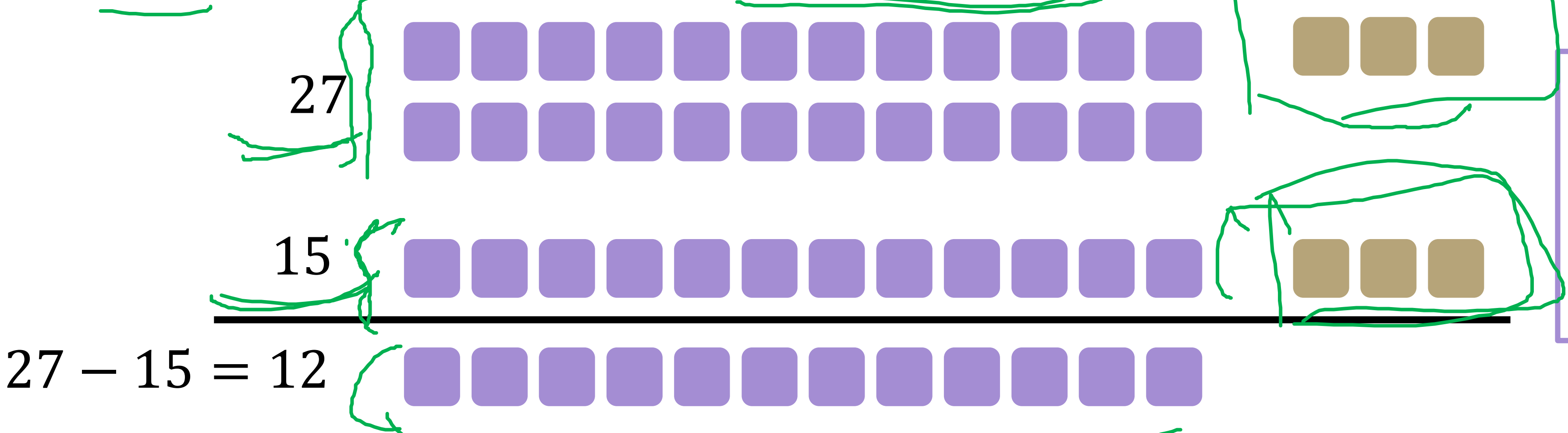
You should be *trying* to do the proof with me. Where do you think we’re going next?

Why?

We'll post an optional (15-minute-ish) video over the weekend with why.

Here's the short version:

It really is equivalent to "what we expected"
`a%n=b%n` if and only if $n|(b-a)$



When you subtract, the remainders cancel. What you're left with is a multiple of 12.

The divides version is much easier to use in proofs…

Claim: for all $a, b, c, n \in \mathbb{Z}, n \geq 0$: $a \equiv b \pmod{n} \rightarrow a + c \equiv b + c \pmod{n}$

Before we start, we must know:

1. What every word in the statement means.
2. What the statement as a whole means.
3. Where to start.
4. What your target is.

Pollev.com/uwcse311

Divides

For integers x, y we say $x|y$ ("x divides y") iff there is an integer z such that $xz = y$.

Equivalence in modular arithmetic

Let $a \in \mathbb{Z}, b \in \mathbb{Z}, n \in \mathbb{Z}$ and $n > 0$.
We say $a \equiv b \ (mod\ n)$ if and only if $n|(b - a)$

Claim: $a, b, c, n \in \mathbb{Z}, n \cancel{\geq} > 0$: $a \equiv b \pmod{n} \rightarrow a + c \equiv b + c \pmod{n}$

Proof:

Let a, b, c, n be arbitrary integers with $n \cancel{\geq} > 0$,
and suppose $a \equiv b \pmod{n}$.

$n \mid b - a + (c - c)$

$\exists z \in \mathbb{Z},\ nz = b - a$

$nz = b + c - (a + c)$

$n \mid (b + c) - (a + c)$

$a + c \equiv b + c \pmod{n}$

Divides

For integers x, y we say $x|y$ ("x divides y") iff there is an integer z such that $xz = y$.

Equivalence in modular arithmetic

Let $a \in \mathbb{Z}, b \in \mathbb{Z}, n \in \mathbb{Z}$ and $n > 0$.
We say $a \equiv b \ (mod\ n)$ if and only if $n|(b - a)$.

A proof

Claim: $a, b, c, n \in \mathbb{Z}, n \geq 0$: $a \equiv b \pmod{n} \rightarrow a + c \equiv b + c \pmod{n}$

Proof:

Let a, b, c, n be arbitrary integers with $n > 0$,
and suppose $a \equiv b (\mathrm{mod}\, n)$.

By definition of mod, $\mathrm{n}|(b - a)$

By definition of divides, $nk = (b - a)$ for some integer k.

Adding and subtracting c, we have $nk = ([b + c] - [\mathrm{a} + \mathrm{c}])$.

Since k is an integer $n|([b + c] - [a + c])$

By definition of mod, $a + c \equiv b + c \pmod{n}$

You Try!

Claim: for all $a, b, c, n \in \mathbb{Z}, n > 0$: If $a \equiv b \ (mod\ n)$ then $ac \equiv bc \ (mod\ n)$

Before we start we must know:

1. What every word in the statement means.
2. What the statement as a whole means.
3. Where to start.
4. What your target is.

Divides

For integers x, y we say $x|y$ ("x divides y") iff there is an integer z such that $xz = y$.

Equivalence in modular arithmetic

Let $a \in \mathbb{Z}, b \in \mathbb{Z}, n \in \mathbb{Z}$ and $n > 0$.
We say $a \equiv b \ (mod\ n)$ if and only if $n|(b - a)$

Claim: for all $a, b, c, n \in \mathbb{Z}, n > 0$: If $a \equiv b\ (mod\ n)$ then $ac \equiv bc\ (mod\ n)$

Proof:

Let a, b, c, n be arbitrary integers with $n > 0$
and suppose $a \equiv b (mod\ n)$.

$ac \equiv bc\ (mod\ n)$

Claim: for all $a, b, c, n \in \mathbb{Z}, n > 0$: If $a \equiv b \ (mod\ n)$ then $ac \equiv bc \ (mod\ n)$

Proof:

Let a, b, c, n be arbitrary integers with $n > 0$
and suppose $a \equiv b (mod\ n)$.

By definition of mod $n | (b - a)$

By definition of divides, $nk = b - a$ for some integer k

Multiplying both sides by c, we have $n(ck) = bc - ac$.

Since c and k are integers, $n | (bc - ac)$ by definition of divides.

So, $ac \equiv bc \ (mod\ n)$, by the definition of mod.

Don’t lose your intuition!

Let’s check that we understand “intuitively” what mod means:

$x \equiv 0 \pmod{2}$

“x is even” Note that negative (even) x values also make this true.

$-1 \equiv 19 \pmod{5}$

This is true! They both have remainder 4 when divided by 5.

$y \equiv 2 \pmod{7}$

This is true as long as $y = 2 + 7k$ for some integer k

Extra Set Practice

Extra Set Practice

Show $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Proof:
Firse, we'll show: $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$
Let x be an arbitrary element of$A \cup (B \cap C)$.
Then by definition of ∪,∩ we have:
$x \in A \vee (x \in B \wedge x \in C)$
Applying the distributive law, we get
$(x \in A \vee x \in B) \wedge (x \in A \vee x \in C)$
Applying the definition of union, we have:
$x \in (A \cup B)$ and $x \in (A \cup C)$
By definition of intersection we have $x \in (A \cup B) \cap (A \cup C)$.
So $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$.

Now we show $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$
Let x be an arbitrary element of $(A \cup B) \cap (A \cup C)$.
By definition of intersection and union, $(x \in A \vee x \in B) \wedge (x \in A \vee x \in C)$
Applying the distributive law, we have $x \in A \vee (x \in B \wedge x \in C)$
Applying the definitions of union and intersection, we have $x \in A \cup (B \cap C)$
So $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$.
Combining the two directions, since both sets are subsets of each other, we have $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Extra Set Practice

Suppose $A \subseteq B$. Show that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$.

Let A, B be arbitrary sets such that $A \subseteq B$.

Let X be an arbitrary element of $\mathcal{P}(A)$.

By definition of powerset, $X \subseteq A$.

Since $X \subseteq A$, every element of X is also in A. And since $A \subseteq B$, we also have that every element of X is also in B.

Thus $X \in \mathcal{P}(B)$ by definition of powerset.

Since an arbitrary element of $\mathcal{P}(A)$ is also in $\mathcal{P}(B)$, we have $\mathcal{P}(A) \subseteq \mathcal{P}(B)$.

Extra Set Practice

Disprove: If $A \subseteq (B \cup C)$ then $A \subseteq B$ or $A \subseteq C$

Consider $A = \{1,2,3\}, B = \{1,2\}, C = \{3,4\}$.

$B \cup C = \{1,2,3,4\}$ so we do have $A \subseteq (B \cup C)$, but $A \nsubseteq B$ and $A \nsubseteq C$.

When you disprove a ∀, you're just providing a counterexample (you're showing ∃) – your proof won't have "let x be an arbitrary element of A."