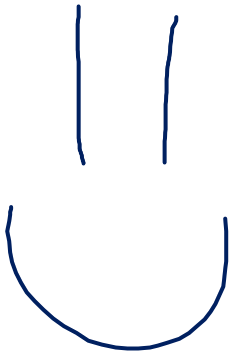


Proof by Contrapositive
Number Theory Definitions

CSE 311 Winter 2024
Lecture 9

Today

Another proof technique (proof by contrapositive)

Start on Number theory definitions

Proof by Contrapositive

Another Proof

Claim: $\forall a(\texttt{Even}(a^2) \rightarrow \texttt{Even}(a))$ "if a^2 is even, then a is even."

See how far you get (this is somewhat a trick question).

At the very least, introduce variables, assume anything you can at the start, put down your "target" at the bottom of the paper.

Trying a direct proof

$\forall a(\texttt{Even}(a^2) \rightarrow \texttt{Even}(a))$ "if a^2 is even, then a is even."

Let a be an arbitrary integer

Suppose a^2 is Even

$a^2 = 2k$, k is an integer.

Therefore a is even.

Trying a direct proof

$\forall a(\texttt{Even}(a^2) \rightarrow \texttt{Even}(a))$

Let a be an arbitrary integer and suppose that a^2 is even.

By definition of even, $a^2 = 2k$ for some integer k.

Taking the positive square-root of each side, we get $a = \sqrt{2k}$

….

Taking a square root of a variable is tricky! It's hard to do algebra on.

Therefore a is even.

Trying a direct proof

$\forall a(\mathtt{Even}(a^2)\rightarrow\mathtt{Even}(a))$

Let a be an arbitrary  s even.

By definition of even

Taking the positive s et $a = \sqrt{2k}$

….

Therefore a is even.

There has to be a better way!

What should we do?

We’re trying to show an implication. How can we transform implications? Could that make it easier?

Maybe a transformation that would “switch the order” so that instead of taking a square root, we’re squaring…

Take a contrapositive!

Proving by contrapositive

$\forall a(\mathtt{Even}(a^2) \rightarrow \mathtt{Even}(a)) \equiv \forall a(\neg\mathtt{Even}(a) \rightarrow \neg\mathtt{Even}(a^2)) \equiv \forall a(\mathtt{Odd}(a) \rightarrow \mathtt{Odd}(a^2))$

We argue by contrapositive.

Let a be an arbitrary integer and suppose a is odd.

we thus get that a^2 meets the definition of odd (being 2 times an integer plus one), as required.

Since a was arbitrary, we have that for every odd a, that a^2 is also odd, which is the contrapositive of our original claim.

Proving by contrapositive

~~Even~~$(a) \to$ Even(a^2)

$\forall a(\text{Even}(a^2) \rightarrow \text{Even}(a)) \equiv \forall a(\neg\text{Even}(a) \rightarrow \neg\text{Even}(a^2)) \equiv \forall a(\text{Odd}(a) \rightarrow \text{Odd}(a^2))$

We argue by contrapositive.

Let a be an arbitrary integer and suppose a is odd.

By definition of odd, $a = 2k + 1$ for some integer k.

Squaring both sides, we get $a^2 = (2k+1)^2 = 4k^2 + 4k + 1$

Rearranging, we get $a^2 = 2(2k^2 + 2k) + 1$. Since k is an integer, $2k^2 + 2k$ is an integer, we thus get that a^2 meets the definition of odd (being 2 times an integer plus one), as required.

Since a was arbitrary, we have that for every odd a, that a^2 is also odd, which is the contrapositive of our original claim.

Proof by contrapositive in general

You might write down the contrapositive for yourself, but it doesn't go in the proof.

Tell your reader you're arguing by contrapositive right at the start! (Otherwise it'll look like you're proving the wrong thing!)

The quantifier(s) don't change! Just the implication inside.

Signs you might want to use proof by contrapositive

1. The hypothesis of the implication you're proving has a "not" in it (that you think is making things difficult)

2. The target of the implication you're proving has an "or" or "not" in it.

3. There's a step that is difficult forward, but easy backwards
e.g., taking a square-root forward, squaring backwards.

4. You get halfway through the proof and you can't "get ahold of" what you're trying to show.
e.g., you're working with a "not equal" instead of an "equals" or "every thing doesn't have this property" instead of "some thing does have that property"

All of these are reasons you **might** want contrapositive. Sometimes you just have to try and see what happens!

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 $n = pq$, $\lambda(n) = \mathrm{lcm}(\lambda(p), \lambda(q))$, 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 $\mathrm{lcm}(a, b) = |ab|/\gcd(a, b)$.
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 $\mathrm{lcm}(a, b)$

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 $0 \le m < n$ 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.

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

$2|6$ $x|y$

Divides

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

$2 \cdot 3 = 6$

"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*" *when both are positive integers

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?

$2\|4$ $2\cdot 2=4$	$4\|2$ $4\cdot\_=2$	$2\|-2$ $2\cdot(-1)=-2$
$5\|0$ $5\cdot 0=0$	$0\|5$	$1\|5$

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

a is an integer

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 $0 \leq r < d$
Such that $a = dq + r$

$$\frac{a}{d}$$

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

$$\begin{array}{r} 4 \text{ R}5 \\ 7\overline{)33} \\ \underline{28} \\ 5 \end{array}$$

q

r

q is the “quotient”
r is the “remainder”

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

$$a = qd + r$$

Unique

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 $0 \leq r < d$
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 $a \in \mathbb{Z}$, $d \in \mathbb{Z}$ with $d > 0$
There exist *unique* integers q, r with $0 \le r < d$
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

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

$\equiv$

To say “the same” we don’t want to use = … that means the normal =

$13 \equiv 1 \pmod{12}$

We’ll write $13 \equiv 1 (\text{mod } 12)$

$\equiv$ 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 $\equiv$ symbol, it’s not operating on 1.

Modular Arithmetic

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 % ?

$13 \% 12 = 1 \% 12$

Modular Arithmetic

$n \mid (b - a)$

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

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!

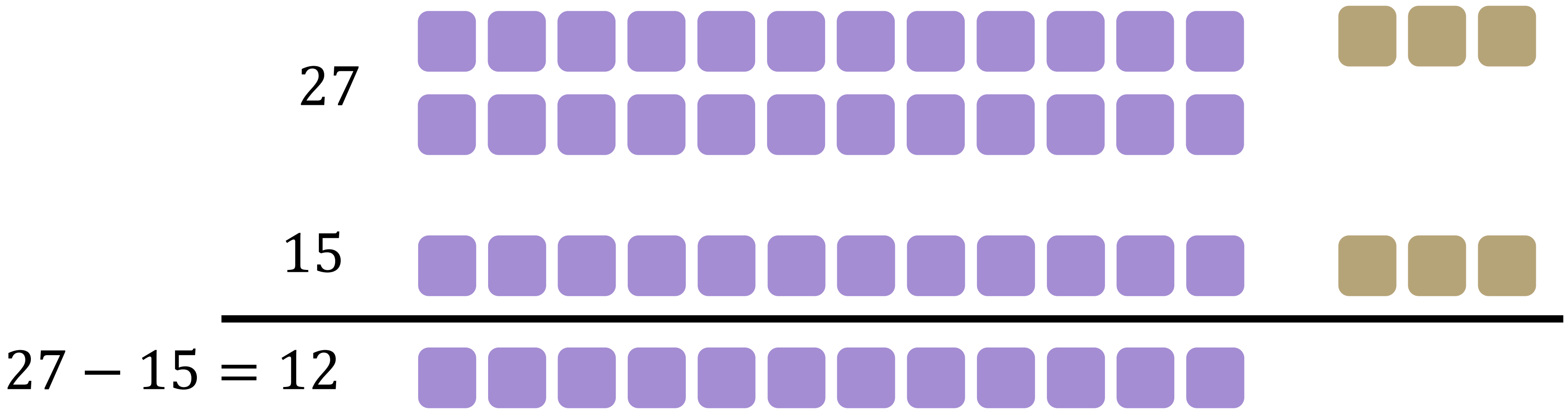
You should be *trying* to do the proof with me. Where do you think we’re going next?

Why?

Your Tas will take a bit of time in section on this.

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…

Another contrapositive example

Another Proof

For all integers, a, b, c: Show that if $a \nmid (bc)$ then $a \nmid b$ or $a \nmid c$.

Proof:

Let a, b, c be arbitrary integers, and suppose $a \nmid (bc)$.

Then there is not an integer z such that $az = bc$

…

So $a \nmid b$ or $a \nmid c$

Another Proof

For all integers, a, b, c: Show that if $a \nmid (bc)$ then $a \nmid b$ or $a \nmid c$.

Proof:

Let a, b, c be arbitrary integers, and suppose $a \nmid (bc)$.

Then there is not an integer z such that $az = bc$

…

So $a \nmid b$ or $a \nmid c$

Another Proof

For all integers, a, b, c: Show that if $a \nmid (bc)$ then $a \nmid b$ or $a \nmid c$.

There has to be a better way!

If only there were some equivalent implication…

One where we could negate everything…

Take the contrapositive of the statement:

For all integers, a, b, c: Show if $a|b$ and $a|c$ then $a|(bc)$.

By contrapositive

Claim: For all integers, a, b, c: Show that if $a \nmid (bc)$ then $a \nmid b$ or $a \nmid c$.

We argue by contrapositive.

Let a, b, c be arbitrary integers, and suppose $a|b$ and $a|c$.

Therefore $a|bc$

By contrapositive

Claim: For all integers, a, b, c: Show that if $a \nmid (bc)$ then $a \nmid b$ or $a \nmid c$.

We argue by contrapositive.

Let a, b, c be arbitrary integers, and suppose $a|b$ and $a|c$.

By definition of divides, $ax = b$ and $ay = c$ for integers x and y.

Multiplying the two equations, we get $axay = bc$

Since a, x, y are all integers, xay is an integer. Applying the definition of divides, we have $a|bc$.