Sketch of Lecture 1

Wed, 1/20/2021

Review: The calculus of congruences

Example 1. Today is Wednesday. What day of the week will it be a year (365 days) from now?

Solution. Since $365 \equiv 1 \pmod{7}$, it will be Thursday on 1/20/2022.

$$a \equiv b \pmod{n} \qquad \text{means} \qquad a = b + mn \quad (\text{for some } m \in \mathbb{Z})$$

In that case, we say that "a is congruent to b modulo n".

In other words: $a \equiv b \pmod{n}$ if and only if $a - b$ is divisible by n.

Example 2. $17 \equiv 5 \pmod{12}$ as well as $17 \equiv 29 \equiv -7 \pmod{12}$

We say that $5, 17, 29, -7$ all represent the same **residue** modulo 12.

There are exactly 12 different residues modulo 12.

Example 3. Every integer x is congruent to one of $0, 1, 2, 3, 4, \ldots, 11$ modulo 12.

We therefore say that $0, 1, 2, 3, 4, \ldots, 11$ form a **complete set of residues** modulo 12.

Another natural complete set of residues modulo 12 is: $0, \pm 1, \pm 2, \ldots, \pm 5, 6$

[-6 is not included because $-6 \equiv 6$ modulo 12.]

Online homework. When entering solutions modulo n for online homework, your answer needs to be from one of the two natural sets of residues above.

Example 4. Modulo 7, we have the complete sets of residues $0, 1, 2, 3, 4, 5, 6$ and $0, \pm 1, \pm 2, \pm 3$. A less obvious set is $0, 3, 3^2, 3^3, 3^4, 3^5, 3^6$.

Review. Note that $3^6 \equiv 1 \pmod{7}$ by **Fermat's little theorem**. Because 6 is the smallest positive exponent such that $3^k \equiv 1 \pmod{7}$, we say that the **multiplicative order** of $3 \pmod{7}$ is 6. This makes $3 \pmod{7}$ a **primitive root**.

On the other hand, the **multiplicative order** of $2 \pmod{7}$ is 3. (Why?!)

Example 5. $67 \cdot 24 \equiv 4 \cdot 3 \equiv 5 \pmod{7}$

The point being that we can (and should!) reduce the factors individually first (to avoid the large number we would get when actually computing $67 \cdot 24$ first). This idea is crucial in the computations we (better, our computers) will later do for cryptography.

Example 6. (but careful!) If $a \equiv b \pmod{n}$, then $ac \equiv bc \pmod{n}$ for all integers c.

However, the converse is not true! We can have $ac \equiv bc \pmod{n}$ without $a \equiv b \pmod{n}$ (even assuming that $c \not\equiv 0$).

For instance. $2 \cdot 4 \equiv 2 \cdot 1 \pmod{6}$ but $4 \not\equiv 1 \pmod{6}$

However. $2 \cdot 4 \equiv 2 \cdot 1 \pmod{6}$ means $2 \cdot 4 = 2 \cdot 1 + 6m$. Hence, $4 = 1 + 3m$, or, $4 \equiv 1 \pmod{3}$.

The issue is that 2 is not invertible modulo 6.

a is invertible modulo n $\iff$ $\gcd(a, n) = 1$

Similarly, $ab \equiv 0 \pmod{n}$ does not always imply that $a \equiv 0 \pmod{n}$ or $b \equiv 0 \pmod{n}$.

For instance. $4 \cdot 15 \equiv 0 \pmod{6}$ but $4 \not\equiv 0 \pmod{6}$ and $15 \not\equiv 0 \pmod{6}$

Good news. These issues do not occur when n is a **prime** p.

- If $ab \equiv 0 \pmod{p}$, then $a \equiv 0 \pmod{p}$ or $b \equiv 0 \pmod{p}$.
- Suppose $c \not\equiv 0 \pmod{p}$. If $ac \equiv bc \pmod{p}$, then $a \equiv b \pmod{p}$.

Example 7. Determine $4^{-1} \pmod{13}$.

Recall. This is asking for the **modular inverse** of 4 modulo 13. That is, a residue x such that $4x \equiv 1 \pmod{13}$.

Brute force solution. We can try the values $0, 1, 2, 3, \ldots, 12$ and find that $x = 10$ is the only solution modulo 13 (because $4 \cdot 10 \equiv 1 \pmod{13}$).

This approach may be fine for small examples when working by hand, but is not practical for serious congruences. On the other hand, the Euclidean algorithm, reviewed below, can compute modular inverses extremely efficiently.

Glancing. In this special case, we can actually see the solution if we notice that $4 \cdot 3 = 12$, so that $4 \cdot 3 \equiv -1 \pmod{13}$ and therefore $4^{-1} \equiv -3 \pmod{13}$.

Example 8. Solve $4x \equiv 5 \pmod{13}$.

Solution. From the previous problem, we know that $4^{-1} \equiv -3 \pmod{13}$.
Hence, $x \equiv 4^{-1} \cdot 5 \equiv -3 \cdot 5 = -2 \pmod{13}$.

(Bézout's identity) Let $a, b \in \mathbb{Z}$ (not both zero). There exist $x, y \in \mathbb{Z}$ such that

$$\gcd(a, b) = ax + by.$$

The integers x, y can be found using the **extended Euclidean algorithm**.
In particular, if $\gcd(a, b) = 1$, then $a^{-1} \equiv x \pmod{b}$.

Here, $\mathbb{Z}$ denotes the set of all integers $0, \pm 1, \pm 2, \ldots$

Example 9. Find $d = \gcd(17, 23)$ as well as integers r, s such that $d = 17r + 23s$.

Solution. We apply the extended Euclidean algorithm:

$$\begin{array}{lll} \gcd(17, 23) & \boxed{23} = 1 \cdot \boxed{17} + 6 & \text{or:} \quad \boxed{A} \quad 6 = 1 \cdot \boxed{23} - 1 \cdot \boxed{17} \\ = \gcd(6, 17) & \boxed{17} = 3 \cdot \boxed{6} - 1 & \qquad\;\; \boxed{B} \quad 1 = -1 \cdot \boxed{17} + 3 \cdot \boxed{6} \\ = 1 & & \end{array}$$

Backtracking through this, we find that:

$$1 \underset{\boxed{B}}{=} -1 \cdot \boxed{17} + 3 \cdot \boxed{6} \underset{\boxed{A}}{=} -4 \cdot \boxed{17} + 3 \cdot \boxed{23}$$

That is, **Bézout's identity** takes the form $1 = -4 \cdot 17 + 3 \cdot 23$.

Example 10. Determine $17^{-1} \pmod{23}$.

Solution. By the previous example, $1 = -4 \cdot 17 + 3 \cdot 23$. Reducing modulo 23, we get $-4 \cdot 17 \equiv 1 \pmod{23}$.
Hence, $17^{-1} \equiv -4 \pmod{23}$. [Or, if preferred, $17^{-1} \equiv 19 \pmod{23}$.]

Example 11. Determine $16^{-1}\pmod{25}$.

Solution. We apply the extended Euclidean algorithm:

$$\begin{array}{llll} \gcd(16,25) & \boxed{25}=2\cdot\boxed{16}-7 & \text{or:}\ \boxed{A} & 7=-1\cdot\boxed{25}+2\cdot\boxed{16} \\ =\gcd(7,16) & \boxed{16}=2\cdot\boxed{7}+2 & \phantom{\text{or:}}\ \boxed{B} & 2=1\cdot\boxed{16}-2\cdot\boxed{7} \\ =\gcd(2,7) & \boxed{7}=3\cdot\boxed{2}+1 & \phantom{\text{or:}}\ \boxed{C} & 1=\boxed{7}-3\cdot\boxed{2} \\ =1 & & & \end{array}$$

Backtracking through this, we find that:

$$1 \underset{\boxed{C}}{=} \boxed{7}-3\cdot\boxed{2} \underset{\boxed{B}}{=} 7\cdot\boxed{7}-3\cdot\boxed{16} \underset{\boxed{A}}{=} -7\cdot\boxed{25}+11\cdot\boxed{16}$$

That is, **Bézout's identity** takes the form $-7\cdot 25+11\cdot 16=1$.

Reducing modulo 25, we get $11\cdot 16\equiv 1\pmod{25}$. Hence, $16^{-1}\equiv 11\pmod{25}$.

Application: credit card numbers

Have you ever thought about the numbers on your credit card? Usually, these are 16 digits. For instance, 4266 8342 8412 9270.

No worries (or false hopes...). While close, this is not exactly my credit card number.

- The first digit(s) of a credit card identify the issuer of the card. For instance, a leading 4 is typically Visa, 51 to 55 indicate Mastercard, and 34, 37 indicate American Express. The above credit card is indeed a Visa card.

  More information at: https://en.wikipedia.org/wiki/Payment_card_number

- The last digit is a **check digit**, and a valid credit card number must pass the **Luhn check** (patented by IBM scientist Hans Peter Luhn in 1954/60; now in public domain).

  This works as follows: every second digit, starting with the first, is doubled. If that results in a two-digit number, we take the sum of those two digits.

  $$\left[\begin{array}{ccccccccccccccccc} & 4 & 2 & 6 & 6 & 8 & 3 & 4 & 2 & 8 & 4 & 1 & 2 & 9 & 2 & 7 & 0 \\ \times 2 & 8 & & 12 & & 16 & & 8 & & 16 & & 2 & & 18 & & 14 & \\ & 8 & 2 & 3 & 6 & 7 & 3 & 8 & 2 & 7 & 4 & 2 & 2 & 9 & 2 & 5 & 0 \end{array}\right]$$

  The other half of the digits is left unchanged. We then add all these digits and reduce modulo 10:

  $$8+2+3+6+7+3+8+2+7+4+2+2+9+2+5+0\equiv 0\pmod{10}$$

  The result of that computation must be 0. Otherwise, the credit card number fails the Luhn check and is invalid.

Example 12. (extra exercise)

(a) Check that the number 4266 8342 8412 9280 fails the Luhn check.

(b) How do we have to change the last digit to turn this into a valid credit card number?

The purpose of the Luhn check is to detect accidental errors.

[A random credit card number has a 90% chance of failing the Luhn check. Why?!]

On the other hand, as the previous example shows, it provides basically no protection against malicious attacks (except against amateur criminals not aware of the Luhn check).

The Luhn check was designed before online banking (patent filed in 1954). So a special focus is on detecting accidental errors that occur frequently when writing down (things like) credit card numbers by hand.

- For instance, it is common that a single digit gets messed up. Every such error is detected by the Luhn check. (Why?!)
- Another common error is to transpose two digits. Every such error (with the exception of 09 versus 90) is detected.

  For instance. A 82 at the beginning contributes $7+2=9$ to the check sum, while a 28 contributes $4+8\equiv 2$ to the sum. Hence, replacing one with the other will result in the Luhn check failing.

  Advanced comment. An alternative checksum formula that can detect all single digit changes as well as all transpositions is the Verhoeff algorithm (1969). It is, however, much more complicated and cannot be readily performed by hand.

Example 13. The doubling and sum-of-digits procedure permutes the digits as follows:

original digit	0	1	2	3	4	5	6	7	8	9
adjusted digit	0	2	4	6	8	1	3	5	7	9
difference (mod 10)	0	1	2	3	4	6	7	8	9	0

Note. Looking at the differences modulo 10, we can see why the Luhn check is able to detect all transposition errors (except 09 versus 90).

Example 14. The Luhn check has the somewhat complicated feature that every second digit has to be doubled. Why do we not just add all the original digits and reduce the sum modulo 10?

Solution. One reason is that this simplified check does not catch the transposition of two digits. Why?!

[On the other hand, that simplified check does also detect if just a single digit is incorrect.]

Example 15. (extra) The International Standard Book Number ISBN-10 consists of nine digits $a_1 a_2 \ldots a_9$ followed by a tenth check digit a_{10} (the symbol X is used if the digit equals 10), which satisfies

$$a_{10} \equiv \sum_{k=1}^{9} k a_k \pmod{11}.$$

The ISBN 0-13-186239-? is missing the check digit (printed as "?"). Compute it!

Solution. $1\cdot 0+2\cdot 1+3\cdot 3+4\cdot 1+5\cdot 8+6\cdot 6+7\cdot 2+8\cdot 3+9\cdot 9=210\equiv 1 \pmod{11}$

Hence, the full ISBN is 0-13-186239-1.

This is another example of **error checking**, which is standard practice for all sorts of identification numbers (such as bank account numbers, VIN). With a little more effort **error correction** is also possible.

Comment. The check digit is designed so that it is always possible to detect when a single digit is messed up. It is also always possible to detect when two digits are transposed.

Sketch of Lecture 3

Mon, 1/25/2021

Euler's phi function

Definition 16. **Euler's phi function** $\phi(n)$ denotes the number of integers in $\{1, 2, \ldots, n\}$ that are relatively prime to n.

In other words, $\phi(n)$ counts how many residues are invertible modulo n.

If the prime factorization of n is $n = p_1^{k_1} \cdots p_r^{k_r}$, then $\phi(n) = n\left(1 - \frac{1}{p_1}\right) \cdots \left(1 - \frac{1}{p_r}\right)$.

Why is this true?

- Why is the formula "obvious" if $n = p^k$ is a prime power?
- On the other hand, for composite n, say $n = ab$, we have: $\boxed{\phi(ab) = \phi(a)\phi(b) \text{ if } \gcd(a, b) = 1}$

  This is a consequence of the Chinese remainder theorem. (Review if necessary! We'll use it later but will only review it briefly then.)

The above formula follows from combining these two observations. Can you fill in the details?

Example 17. Compute $\phi(35)$.

Solution. $\phi(35) = \phi(5 \cdot 7) = \phi(5)\phi(7) = 4 \cdot 6 = 24$

Example 18. Compute $\phi(100)$.

Solution. $\phi(100) = \phi(2^2 \cdot 5^2) = \phi(2^2)\phi(5^2) = (2^2 - 2^1) \cdot (5^2 - 5^1) = 40$

[Alternatively: $\phi(100) = \phi(2^2 \cdot 5^2) = 100\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right) = 40$]

Historical examples of symmetric encryption

Alice wants to send a secret message to Bob.

What Alice sends will be transmitted through an unsecure medium (like the internet), meaning that others can read it. However, it is important to Alice and Bob that noone else can understand it.

The original message is referred to as the **plaintext** m. What Alice actually sends is called the **ciphertext** c (the encrypted message).

Symmetric encryption algorithms rely on a secret key k (from some **key space**) shared by Alice and Bob (but unknown to anyone else).

$$\xrightarrow{m} \underset{\text{secret key: } k}{\overset{\text{Alice}}{\boxed{E\text{: Encrypt}}}} \xrightarrow{E_k(m) = c} \quad c \text{ is sent} \quad \xrightarrow{c} \underset{\text{secret key: } k}{\overset{\text{Bob}}{\boxed{D\text{: Decrypt}}}} \xrightarrow{D_k(c) = m}$$

Our ultimate goal will be to secure messaging against both:

- eavesdropping (goal: **confidentiality**)
- tampering (goal: **integrity** and, even stronger, **authenticity**)

  The symmetric encryption approach, by itself, cannot fully protect against tampering. For instance, an attacker can collect previously sent messages, resend them, or use them to replace new messages. (You could preface each message with something like a time stamp to address these issues. But that's getting ahead of ourselves; and there are better ways.)

Shift cipher

The alphabet for our messages will be $A, B, \ldots, Z$, which we will identify with $0, 1, \ldots, 25$.

So, for instance, C is identified with the number 2.

Example 19. (shift cipher) A key is an integer $k \in \{0, 1, \ldots, 25\}$. Encryption works character by character using

$$E_k\colon \quad x \mapsto x + k \pmod{26}.$$

Obviously, the decryption D_k works as $x \mapsto x - k \pmod{26}$.

The **key space** is $\{0, 1, \ldots, 25\}$. It has size 26. [Well, $k = 0$ is a terrible key. Maybe we should exclude it.]

For instance. If $k = 1$, then the message $HELLO$ is encrypted as $IFMMP$.

If $k = 2$, then the message $HELLO$ is encrypted as $JGNNQ$.

Historic comment. Caesar encrypted some private messages with a shift cipher (typically using $k = 3$). The shift cipher is therefore also often called Caesar's cipher.

While completely insecure today, it was fairly secure at the time (with many of his enemies being illiterate).

Modern comment. Many message boards on the internet "encrypt" things like spoilers or solutions using a shift cipher with $k = 13$. This is called ROT13. What's special about the choice $k = 13$?

Solution. Since $-13 \equiv 13 \pmod{26}$, for ROT13, encryption and decryption are the same!

Example 20. (affine cipher) A slight upgrade to the shift cipher, we encrypt each character as

$$E_{(a,b)}\colon \quad x \mapsto ax + b \pmod{26}.$$

How does the decryption work? How large is the key space?

Solution. Each character x is decrypted via $x \mapsto a^{-1}(x - b) \pmod{26}$.

The key is $k = (a, b)$. Since a has to be invertible modulo 26, there are $\phi(26) = \phi(2) \cdot \phi(13) = 12$ possibilities for a. There are 26 possibilities for b. Hence, the key space has size $12 \cdot 26 = 312$.

Vigenere cipher (vector shift cipher)

See Section 2.3 of our book for a full description of the Vigenere cipher.

This cipher was long believed by many (until early 20th) to be secure against ciphertext only attacks (more on the classification of attacks shortly).

Example 21. Let us encrypt $HOLIDAY$ using a Vigenere cipher with key BAD (i.e. $1, 0, 3$).

	H	O	L	I	D	A	Y
$+$	B	A	D	B	A	D	B
$=$	I	O	O	J	D	D	Z

Hence, the ciphertext is $IOOJDDZ$.

An encrypted message

Example 22. (bonus challenge!) You find a post-it with the following message:

$$TERRGVATF \quad FGENATRE$$

Can you make any sense of it?

(To collect a bonus point, send me an email before next week with the plaintext and how you found it.)