Sets

Set theory → set is a well defined objects

{1, 2, 3}

- cartesian product → $A \times B$ → all possible ordered set

{1, 2} {3, 4} → {[(1,3) (1,4) (2,3) (2,4)]}

Set relations

↓

taking a subset of the cartesian product

if $f(x) = x^2$

2 is related to 4

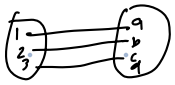
3 is related to 9

- injective : one ↔ to - one

$f: A \to B$ is injective if every $a_1, a_2 \in A$,

if $f(a_1) = f(a_2) \to a_1 = a_2$

also $f(a_1) \neq f(a_2) \to a_1 \neq a_2$



- surjective : onto function

$f: A \to B$, the range of function is equal to its co domain

- bijective → both injective or surjective

Groups :

group is a set with

- an associated and closed binary operator
- an identity element
- every element has an inverse

"abelian group"

- the binary operator is commutive

order of group?

↳ number of elements in the group

"cyclic group"

↳ where elements can generate all other elements by binary operator to its element or inverse.

↳ if group is cyclic → it is abelian group too (reverse not true)

→ group homomorphism

→ product of groups

Rings: is a set with two binary operators (usually add, mul) $+$ $\cdot$

satisfying $a, b, c \in R$

(1) - under first operator, set is abelian group ($R, +$ is abelian)

- closure: $a+b \in R$
- associative: $(a+b)+c = a+(b+c)$
- commutative: $a+b = b+a$
- addition identity: 0
- additive inverse: $a, (-a) \in R$

(2) - under second operator, the set is monoid (semi group) → only inverse property missing to make it group

- closure: $a.b \in R$
- associative: $(a.b).c = a.(b.c) \in R$

(3) - second binary operator over first (distributive law linking $+, \cdot$)

- left distributivity $a.(b+c) = (a.b)+(a.c)$
- right distributivity $(b+c).a = (b.a)+(c.a)$

Examples

→ trivial zero ring ($\{0\}$)
(contains only additive identity $(0+0=0)$ & mul (0.0)

→ Integers ($\mathbb{Z}$)

→ Rational numbers ($\mathbb{Q}$)

Field: field is commutative ring with unity (second operator (mul) has identity '1')

where every non zero element has a "multiplicative inverse"

1 — its a ring so → set of abelian group
- (closure, associative, commutative, additive identity, additive inverse)

2 — Commutative multiplication $a.b = b.a$

3 → multiplication identity $a.1 = 1.a = a$

* 4 → multiplicative inverse $aa^{-1} = a^{-1}a = 1$ (non zero)

Examples

- Rational numbers ($\mathbb{Q}$)
  ↳ add, mul all in $\in \mathbb{Q}$

→ Real numbers

tldr:
- Rings → sets where can add, sub, mul following rules like associativity & distributivity
- Fields → rings where can also always divide by a non zero element
- Finite Fields → a field that contains finite number of elements
  ↓
  "Galois Field"

Excercise 1 –

① Find element congruent to these values: (in finite field)

(i) $-1 \rightarrow 70$
(ii) $-4 \rightarrow 67$
(iii) $-160 \rightarrow 53$
(iv) $500 \rightarrow 3$

② Find element congruent to $a = 5/6$ $b = 11/12$, $c = 21/12$
verify answer by checking $a + b = c$ (in finite field)

$a = 5 \times \frac{1}{6} = 5 \times pow(6, \overset{(-1)}{69}, 71) \equiv 5 \times 12 \equiv 60$

$b = 11 \times \frac{1}{12} = 11 \times pow(12, -1, 71) = 11 \times 6 = 66$

$c = 21 \times \frac{1}{12} = 21 \times 6 = 126$

$60 + 66 = 126$

③ Find element congruent to $a = 2/3$, $b = 1/2$, $c = 1/3$
verify by checking $a \times b = c$ (in field)

$a = 2 \times \frac{1}{3} = 2 \times pow(3, -1, 71) = 48$

$b = pow(2, -1, 71) = 36$

$c = pow(3, -1, 71) = 24$

$48 \times 36 \equiv 24$ (1728)

④ Find inverse of matrix and identity matrix

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix} \quad \det = 4 - 1 = 3 \quad A^{-1} = \frac{1}{3}\begin{bmatrix} 4 & -1 \\ -1 & 1 \end{bmatrix}$$

$$A A^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix} \times \frac{1}{3}\begin{bmatrix} 4 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 4-1 & -1+1 \\ 4-4 & -1+4 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 3 & 0 \\ 0 & +3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

⑤ What is modular square root of 12?
verify by checking $x^2 = 12 \pmod{71}$ → use brute force to find answer in python

$(\sqrt{12}) \bmod 71$ $\quad 2 \times (\sqrt{3}) \bmod 71$ $\quad \sqrt{3} \times \sqrt{3} = 3$

$6 \times \frac{1}{\sqrt{3}}$

```
for x in range(71):
    if x*x % 71 == 12:
        print(x)
```
→ 15, 56

---

* Fermat little theorem

$a^p = a \bmod p$

$a^{p-1} = 1$

$a \, a^{-2} = 1$
$a \, a^{-1} = 1$

$a^{-1} = a^{p-2}$

python – pow(a, -1, p)
pow(a, p-2, p)

for finding square root inverse → brute force
↓
optimize → tonelli shanks algo

⑥ $p(x) = 52x^2 + 24x + 61$, $q(x) = 40x^2 + 40x + 58$ — Find $p(x) + q(x)$, what is $p(x) \times q(x)$. Use galois lib to find roots of $p(x)$ & $q(x)$

$52x^2 + 24x + 61 = 0$

$x = \frac{-24 \pm \sqrt{24^2 - 4 \times 52 \times 61}}{2 \times 52}$

↳ $x = \frac{-24 \pm \sqrt{-48}}{33} \bmod(71)$ — $\sqrt{34}$ ✗

↳ $x = (-24 \pm \sqrt{-48}) \cdot \frac{1}{33} \bmod(71)$ — $\frac{1}{33} \to 28$

$30, 41$

$(-24 \pm 30) \times 28$, $(-24 \pm 41) \times 28$

$-6 \times 28 = 168 \to 26$; $-54 \times 28 = -1512 \to 47$; $17 \cdot 28 = 476 \to 50$; $-65 \cdot 28 = -1820 \to 26$

Pythons:

```
gf = galois.GF(71)
x = galois.Poly([52, 24, 61], field=gf)
x.roots() → GF([39, 42], order=71)
```

$q(x) \to$ no root

$p(x) + q(x) = 92x^2 + 64x + 119 \equiv (21x^2 + 64x + 48) \bmod 71$

$p(x) \cdot q(x) =$ galois → $21x^4 + 58x^3 + 26x^2 + 69x + 59 \to 39, 42$

⑦ $y = ax + b$ — $(10, 15)$, $(23, 29)$

$10a + b = 15$ $\quad 23a + b = 29$

$10a + (-23a + 29) = 15$

$a = +\frac{14}{13}$ $\quad b = -\frac{23 \times 14}{13} - 29 \bmod = \frac{55}{13}$

$f(x) = \frac{14}{13}x + \frac{55}{13}$

$f(10) = 15$

$\frac{140 + 55}{13} = \frac{195}{13} = 15$ ✓

Exercise 2:

① Let set $\mathbb{R}$, show binary is not closed

$a \cdot b \notin \mathbb{R}$

$a \cdot b = \sqrt{a \cdot b}$ → binary operator

$-1 \cdot 2 = \sqrt{-2}$ → complex

② Find binary operator that is closed but not associative $\in \mathbb{R}$

$(a \cdot b) \cdot c \neq a \cdot (b \cdot c)$ $\quad a \cdot b = a - b$

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

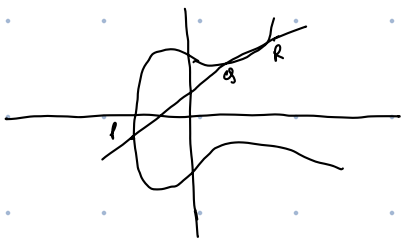
$a - b - k \neq a - b + c$ ✗

③ What algebric group is

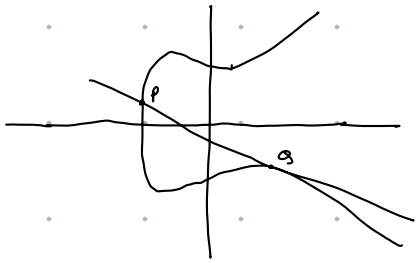
Eliptic curves: $E: y^2 = x^3 + ax + b$, with point at infinity O.

here a, b must satisfy $4a^3 + 27b^2 \neq 0$

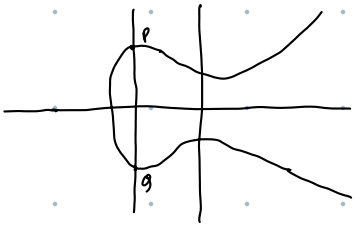
ECC relies on hardness of finding n such that $Q = nP$ given Q and P



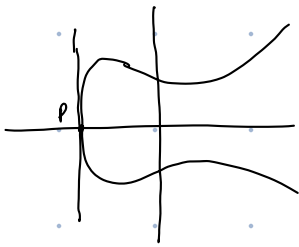
$P + Q + R = O$



$P + Q + Q = O$



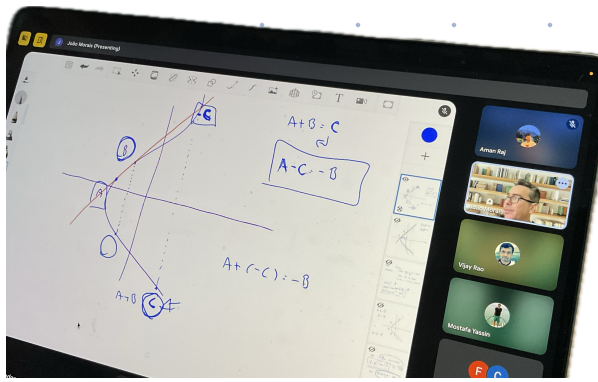
$P + Q + O = O$



$P + P + O = O$

Eliptic curve cryptography is studied over a finite field $\mathbb{F}_p$, it is not a curve, but rather collections of x, y coordinates in $\mathbb{F}_p$

$E(\mathbb{F}_p) = (x, y) : x, y \in \mathbb{F}_p$ satisfying $y^2 = x^3 + ax + b$ $\vee$ $\phi \rightarrow$ infinity



We don't take the three solution of the curve because $(A + B + C = 0)$

$A + B = -C$

associativity will fail

Lets take secp256k1, $a=0$, $b=7$; $P = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$

$y^2 = x^3 + 7$

Excercise → Lets take k1 curve and try ecdsa

py-ecc → BN128 curve → pairing friendly

py-ecc.bn128 → G1 → generator point for one of the group

add → method to do ecc point addition

mul → scalar point multiplication

curve_order → how many points can be generated by adding G1 to itself

neg → method to find negative inverse

All the points on elliptic curves behave like numbers in groups

operation → point addition P+Q, mull K×P → 3P = P+P+P

⇒ Elliptic curve discrete logarithmic problem (ECDLP)

Choosing a point $P = d \times G_1$, very hard to guess d

$\frac{1}{2} G_1 =$ $2^{-1} \bmod n$ (n = curve order)

can verify $P_{inv} + P_m = G$

— Primes → why? → homomorphism

↳ curve_order is also prime

— G1 vs G2 | G1 → generator point on one curve (subgroup) → cordinate (x, y)

G2 → generator point on another related curve → 4 numbers

G2 extension field

F_{p^2} for BN128 G2

new field whose coefficient comes from F_p

$ax+b$, $b+ax$

— we can combine operations result = add(multiply(G1, 2), multiply(G1, 10))

$= 12 G_1$

— solidity precompiles ↳ written in native → BN254 (alt

- ECADD 0x06
- ECMUL 0x 07
- EC pairing 0x08

- Simple static call
- abi encode send calldata
- return bool or return data bytes

add is $O(1)$

mul is $\log_2(N)$

— Introduction of ZK with EC points

Ex → pubkz → $x + y = 8$

Prover knows $x=3$, $y=5$ → Compute $P_x = 3G_2$ $P_y = 5G_1$

verifier → add$(P_x, P_y) = P_{sum}$

mul$(8, G_i) == P_{sum}$ ✓

Now linear equations → $3x + 5y + 6z = 8$

Prover sends P_x P_y P_z

xG_1, yG_1, zG_1

verifier checks $3 \cdot (xG_2) + 5 \cdot (yG_2) + 6 \cdot (zG_1) = 8G_2$

↳ works as $k(aG) = kaG$

↓

this only works on addition $x \cdot y = q$

Prov → xG_1, yG_1

verifier xyG_1 → this is where pairing comes → for multiplication

Recap → field is where you can add, mul, sub, and divide

these operations can also be defined for polynomials

An extension field ($\mathbb{F}_{p^2}$ or $\mathbb{F}_{p^k}$) is field whose elements are polynomial of degree $< k$

G2 (4 numbers) → Point on elliptic curve (x, y)

for G1, x,y are elements of $\mathbb{F}_p$ (single numbers)

for G2, x,y are elements of $\mathbb{F}_{p^2}$ (a polynomial like $ax+b$, $a,b \in \mathbb{F}_p$)

Now, we know we can prove linear equations with EC add, for mul we can use pairing

Pairings: It is not multiplication of two points directly.

3 groups involved:

- G1: An additive group of EC points on BN128 curve.

$$aG_1 + bG_1 = (a+b)G_1$$

- G2: Another additive group of EC points on BN128 twisted curve

$$aG_2 + bG_2 = (a+b)G_2$$

- GT (or G12): A multiplicative group, elements are not typical EC points rather extension field much larger $\mathbb{F}_{p^{12}}$. Denote generator g_T

$$g_T^a \cdot g_T^b = g_T^{a+b}$$

Bilinear map → a pairing is bilinear map if $e: G_1 \times G_2 \to G_T$

- It takes input of one element of G1, one of G2, output one of GT

- The bilinear property (crucial) ***

for any points $P \in G_1$, $Q \in G_2$ & a, b are scalars

$$e(aP, bQ) = e(P, Q)^{ab}$$

↓ if we use generators G_1, h_2, $G_T = e(G_1, G_2)$

$$e(a G_1, b G_2) = (e(G_1, G_2))^{ab} = g_T^{ab}$$

scalar moves out

Now lets take $n_y \geq 12$, $x=3, y=4$

prover → $P_x = 3G_1$ $P_y = 4G_2$ (Note, one in G_1, other in G_2)

verifier → have P_x, P_y, eqn

$LHS = e(P_x, P_y) = e[3G_1, 4G_2) = g_T^{xy} = g_T^{12}$

$RHS =$ directly calculate g_T^{12}

Solidity pairing pre compile (0x08)

↳ doesn't return the GT element $e(P, Q)$

↳ returns sums of pairings equal identity in GT

↳ takes a list of concatenated $(P G_1, Q G_2)$ pairs

Lets take some example $x=3, y=12$

$$e(3G_1, 4G_2) = e(G_1, 12G_2) = e(12G_1, G_2)$$

↓ this is equivalent to check

$$e(3G_1, 4G_2) \cdot e(G_1, -12G_2) = 1_{GT}$$

Input → $3G_1 \rightarrow (x, y)$ ← A

$4G_2 \rightarrow (x_0, x_1, y_0, y_1)$ → Y inverted ← B

$G_1 \rightarrow (x, y)$ generator ← C

$-12G_2$ → to get this, calculate $12G_2$ ← D

flip over y

$(y_0, y_1 \bmod p)$