CS 5602 Introduction to Cryptography
Lecture 09
Category Theory
Chinese Remainder Theorem

George Markowsky
Computer Science Department
Missouri University of Science & Technology

1

Moving 2/26 Class

- I need to be at a University of Missouri System event on Tuesday 2/26
- Consequently, I will not be here to give the lecture
- Instead I will give that lecture online, Friday 2/22 from 7:30 – 8:45 pm
- The lecture will be recorded and you will be able to access the recording in case you can't connect Friday evening
- More details later this week

2

Category Theory

- This is a very abstract theory, sometime referred to as the theory of "abstract nonsense"
- I have found it a useful way to think about entities of all sorts
- Will give a short introduction to the theory and motivate it with examples

3

Categories

- A category C is a triple (Obj, Morph, °) where
- Obj are the objects of the category
- Morph are the morphisms between members of Obj
- If A, B ∈ Obj, hom_C (A,B), or just hom(A,B) if the category is clear, denotes the morphisms from A to B
- Some people call members of hom, homomorphisms, but morphisms is a bit more general
- If f ∈ hom(A,B), we write f: A → B
- ° is an operator called composition

4

Categories

- A category C is a triple (Obj, Morph, °) where
- ° is an operator called composition and it seeks to put together morphisms
- In general if f ∈ hom(A,B) and g ∈ hom(B,C) then g°f ∈ hom(A,C)
- Note hom(A,B) can be empty if A ≠ B
- In general we get the following picture:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "g\circ f"'] & B \arrow[d, "g"] \\ & C \end{tikzcd}$$

5

Categories

- Additional requirements
- ° is associative, i.e., h°(g°f) = (h°g)°f where h ∈ hom(C,D), g ∈ hom(B,C) and f ∈ hom(A,B)
- For every A ∈ Obj, there is an *identity morphism* in hom(A,A), often denoted by 1_A or id_A, that has the following properties
- For all B ∈ Obj, and f ∈ hom(A,B) and g ∈ hom(B,A), we have f = f°id_A and g = id_A°g

6

Diagram Chasing

- Diagrams play a big part in category theory

$$\begin{tikzcd} A \arrow[r, "id_A"] \arrow[dr, "f"'] & A \arrow[d, "f"] \\ & B \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "g"] \arrow[dr, "g"'] & B \arrow[d, "id_B"] \\ & B \end{tikzcd}$$

7

More on Categories

- Sometimes people write fg instead of f°g if there is no confusion
- Sometimes they will write f(g) and use functional notation
- I will try to use ° consistently at the beginning
- It turns out that the ideas that have been presented to you about injective, surjective and bijective can be generalized to categories
- We will often just write morphisms f, g and g°f without always stating that $f \in hom(A,B)$, $g \in hom(B,C)$ and g°f in hom(A,C)
- You are expected to supply the objects and homs so that everything makes sense
- Categories are all about the morphisms!

8

Types of Morphisms

- An *endomorphism* is any morphism that belongs to hom(A,A), i.e., it is a morphism from an object to itself
- A *monomorphism* (monic morphism) is a morphism f such that for all morphisms g_1 and g_2 if $f°g_1 = f°g_2$ then $g_1 = g_2$ (injection)
  - Equivalent to left cancellation
- An *epimorphism* (epic morphism) is a morphism f such that for all morphisms g_1 and g_2 if $g_1°f= g_2°f$ then $g_1 = g_2$ (surjection)
  - Equivalent to right cancellation
- A *bimorphism* is a morphism that is both a monomorphism and an epimorphism (bijection)
- An *automorphism* is a morphism that is both a bimorphism and a endormorphism, i.e., it is a member of hom(A,A) that is a bimorphism

9

Monomorphism

If we have the diagrams below

$$\begin{tikzcd} X \arrow[r, "g_1"] \arrow[dr, "f\circ g_1"'] & A \arrow[d, "f"] \\ & B \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, "g_2"] \arrow[dr, "f\circ g_2"'] & A \arrow[d, "f"] \\ & B \end{tikzcd}$$

f is a monomorphism iff $f\circ g_1 = f \circ g_2 \Rightarrow g_1 = g_2$

10

Epimorphism

If we have the diagrams below

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "g_1\circ f"'] & B \arrow[d, "g_1"] \\ & X \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "g_2\circ f"'] & B \arrow[d, "g_2"] \\ & B \end{tikzcd}$$

f is a epimorphism iff $g_1\circ f = g_2\circ f \Rightarrow g_1 = g_2$

11

Example: The Category of Sets

- What are the objects?
- All sets
- What are the morphisms?
- Functions between sets
- What is °?
- Regular function composition
- Is it a category?
- Yes! Why?

12

Sets Form A Category

- Function composition is associative
- There is an identity function for sets
- How about a theorem to prove?
- You might wonder how there is anything to prove since we apparently have not done much

13

Theorem 1

- A function (morphism) in the category of sets is a monomorphism iff it is injective
- Proof: We break it into two parts (injective → monomorphism) and (monomorphism → injective)
- Suppose $f : A \to B$ is injective and suppose $g_1 : X \to A$ and $g_2: X \to A$ are such that $f°g_1 = f°g_2$. This means that for all $x \in X$, $f(g_1(x)) = f(g_2(x))$
- Since f is injective, this means that $g_1(x) = g_2(x)\ \forall x$, so $g_1 = g_2$. Thus, f is a monomorphism
- Suppose that f is a monomorphism, but not injective

14

Theorem 1

- If f is not injective, $\exists\ x, y \in A$ such that $x \neq y$ and $f(x) = f(y)$
- Let $S = \{1\}$ and let $g_1: S \to A$ be given by $g_1(1) = x$
- Let $g_2: S \to A$ be given by $g_2(1) = y$
- Clearly $g_1 \neq g_2$
- Note that $f°g_1 = f°g_2$ but $g_1 \neq g_2$ which contradicts the assumption that f is a monomorphism

15

Theorems 2 and 3

- A function in the category of sets is an epimorphism iff it is surjective
- A function in the category of sets is a bimorphism iff it is bijective
- Proof: These will be on the next homework
- Notice that we have generalized the ideas of injective, surjective and bijective in a very general way that has no mention of elements
- Notice the similarity with groups and cancellation laws

16

The Category of Graphs

- The objects are graphs
- Graphs are pairs (V, E) where V is a set of vertices and E is a set of one or two element subsets of V called edges
- What would be the morphisms for this category?
- You would start out with functions between sets of vertices, but what would be helpful if you wanted to acknowledge the graph structures?
- Definition: if $G = (V_1, E_1)$ and $H = (V_2, E_2)$ are graphs, $f \in hom(G,H)$ means that $f: V_1 \to V_2$ such that if $\{a,b\} \in E_1$, then $\{f(a), f(b)\} \in E_2$
- In other words, morphisms in the category of graphs preserve edges

17

The Category of Graphs

- Is this a category?
- Function composition is associative
- The identity function on vertices is the identity morphism in the category
- Proving that the identity function is the identity morphism will be a homework exercise

18

The Category of Directed Graphs

- It is probably not surprising for you to learn that Directed Graphs are naturally a category
- The morphisms are maps between vertices that preserve arrows
- In other words, if $G = (V_1, A_1)$ and $H = (V_2, A_2)$ are digraphs, $f \in \mathrm{hom}(G,H)$ means that $f: V_1 \rightarrow V_2$ such that if $(a,b) \in A_1$, then $(f(a), f(b)) \in A_2$
- Can we say that the Category of Graphs is somehow related to the Category of Digraphs?
- Hold onto that thought for a bit, while we ponder other categories or non-categories

19

More Categories

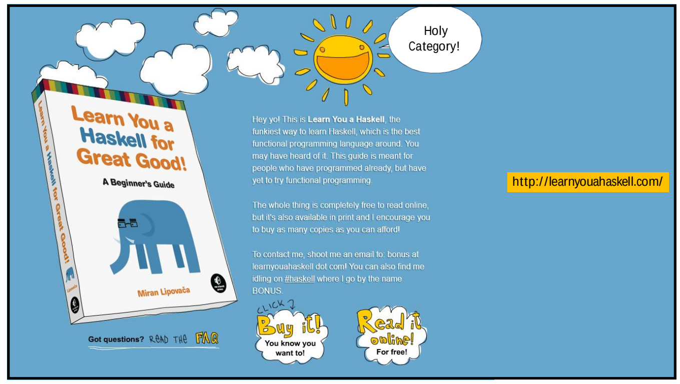
- The category of relations, $R_1(A)$, on a set A. $\mathrm{hom}(R_1,R_2) = \{ f : A \rightarrow A \mid aR_1b \rightarrow f(a)R_2f(b)\}$
- The category of relations, $R_2(A)$, on a set A. $\mathrm{hom}(R_1,R_2) = \{ f : A \rightarrow A \mid aR_1b \text{ iff } f(a)R_2f(b)\}$
- The preceding two categories are different
- It's all about the morphisms!
- Example, let $A = \{ 1, 2, 3\}$, $R_1 = \{ (1,3), (2, 3) \}$, $R_2 = \{(1,1)\}$ and $f: A \rightarrow A$ be given by $f(1) = f(2) = f(3) = 1$. Note that $f \in \mathrm{hom}_{R1}(R_1,R_2)$, but $f \notin \mathrm{hom}_{R2}(R_1,R_2)$ because $f(1)R_2f(2)$ but it is False that $1R_12$.

20

Still More Categories

- The category of partially ordered sets with the morphisms being order preserving maps between them: $a \le b \rightarrow f(a) \le f(b)$
- Many other mathematical objects form categories!
- Groups, rings, fields, vector spaces, etc….
- The programming language Haskell borrows many ideas from Category Theory
- The chief category in Haskell is called Hask
- The objects of Hask are Haskell types and morphisms are Haskell functions – not enough time to present this further

21



22

Theorem

- Suppose in $R_2(A)$, $\mathrm{hom}(R_1,R_2) \neq \emptyset$ and R_2 is an equivalence relation then R_1 is an equivalence relation
- Proof:
- Let $a \in A$. Since R_2 is an equivalence relation, $f(a)R_2f(a)$, so by the definition of $R_2(A)$, we see that aR_1a. Thus, R_1 is reflexive.
- Let $a, b \in A$ and suppose aR_1b. We know that $f(a)R_2f(b)$, but since R_2 is an equivalence relation, $f(b)R_2f(a)$ and therefore bR_2a. Thus, R_1 is symmetric.
- Let $a, b, c \in A$ and suppose aR_1b and bR_1c. It follows that $f(a)R_2f(b)$ and $f(b)R_2f(c)$. Since R_2 is an equivalence relation, $f(a)R_2f(c)$, whence aR_1c. thus, R_1 is transitive. We now see that R_1 is an equivalence relation

23

Functors

- Before we noticed that we had two similar categories, Graphs and Digraphs
- It would be nice to have some notion of when categories are related
- This is the idea of a *functor*
- Let A and B be categories, $F: A \rightarrow B$ is called a functor if for each object X in A, F(X) is an object in B
- For each morphism $f \in \mathrm{hom}(X,Y)$, $F(f) \in \mathrm{hom}(F(X), F(Y))$ such that
- $F(id_X) = id_{F(X)}$ and
- $F(g°f) = F(g)° F(f)$

24

24

Functor from Graphs to Digraphs

- For each G = (V, E) associate the digraph D = (V, A) where (a,b) ∈ A iff {a, b} in E
- Note that this adds two arrows for each edge in G except for self-loops in G which give rise to just a self-loop in D
- You would have to show that the identity morphisms are preserved and composition is preserved, but this is pretty straightforward

25

Functor from Digraphs to Graphs

- For each digraph D = (V, A) define a graph G = (V, E) where {a,b} in E iff either (a,b) ∈ A or (b,a) ∈ A
- Note that this basically takes the direction off the arrows
- If you have arrows (a,b) and (b,a) you still only get one edge {a, b}

26

Functor from Graphs to Digraphs

- What might this look like?
- I will ask you this question on the next homework

27

More Functors

- Let Group be the category of groups where Hom(G_1,G_2) consists of all group homomorphisms from $G_1 \rightarrow G_2$
- Note that Hom(G_1,G_2) ≠ ∅ because you always have the trivial homorphism where f(g) = 1 ∀ g ∈ G_1
- Let Ring be the category of rings with morphisms being ring homomorphisms
- Let Field be the category of fields with morphisms being field homomorphisms
- We have all sorts of functors available, for example RG: Ring → Group where RG(R) is the (R,+) group
- Consider FG: Field → Group where FG(F) = (F – {0},*)
- etc.

28

Applications to Computer Science

- Very important in understanding the semantics of programming languages
- People are very concerned about scaling up programs without errors
- Imperative languages have side-effects which lead to bugs and security holes
- Functional programming languages seek to secure as much of the code as possible
- Much of this work is very abstract and mathematical
- A key feature of the Haskell Programming Language

29

Category Theory for Computing Science

Michael Barr
Charles Wells

http://www.math.mcgill.ca/triples/Barr-Wells-ctcs.pdf

30

Category Theory

A Gentle Introduction

Peter Smith

University of Cambridge

http://www.logicmatters.net/resources/pdfs/GentleIntro

A Gentle Introduction to Category Theory

— the calculational approach —

Maarten M. Fokkinga

https://ris.utwente.nl/ws/portalfiles/portal/6141835

31

Studying $\mathbb{Z}_n$

- You have already gotten the notion that integers modulo n will be very important in cryptography
- We will now begin a deeper look at $\mathbb{Z}_n$ and its very interesting properties
- We know that $\mathbb{Z}_n$ is an additive abelian group, but $\mathbb{Z}_n - \{0\}$ is only a multiplicative group if n is a prime
- We know that the elements $q \in \mathbb{Z}_n - \{0\}$ that are relatively prime to n form an abelian, multiplicative group and that this is the largest subset of $\mathbb{Z}_n - \{0\}$ that is a multiplicative subgroup
- If p and q are integers such that GCD(p,q) = 1 we say that p and q are coprime or relatively prime

32

Equations mod n

- Suppose we have equations of the form a*x = b (mod n)
- Here are some questions
- Can we always find at least one solution for the above equation?
- Can there be more than one solution for such an equation?
- If there can be more than one solution, how many solutions can there be?
- Try some equations
- 7x = 3 (mod 143)
- 2x = 3 (mod 8)
- 11x = 22 (mod 143)

33

Equations

- 7x = 3 (mod 143) has the unique solution x = 123
- Where did that come from?
- 2x = 3 (mod 8) has no solution because 2x is always even and can never have a remainder of 3 mod 8
- 11x = 22 (mod 143) has the following 11 solutions: 2, 15, 28, 41, 54, 67, 80, 93, 106, 119, and 132 (do you see a pattern?)
- Let's approach this systematically

34

Solving ax = b (mod n)

- Note that if ax = b (mod n) this means that ∃ c such that ax + cn = b
- Note that if g = GCD(a,n), then (ax + cn) % g = 0
- In particular, there is no solution if b%g ≠ 0
- Recall the equations
- 7x = 3 (mod 143), GCD(7,143) = 1 and 3%1 = 0
- 2x = 3 (mod 8), GCD(2,8) = 2 and 3%2 ≠ 0
- 11x = 22 (mod 143), GCD(11,143) = 11 and 22%11 = 0
- Is b%g = 0 sufficient for there to be a solution of ax = b (mod n)?

35

Solving ax = b (mod n)

- Suppose b%g = 0, then b = k*g
- Note that from the extended GCD algorithm, ∃ s and t such that g = s*a +t*n, which means that k*s*a + k*t*n = k*g (mod n) which means that k*s*a = b (mod n)
- Applying this to the equation 7X = 3 (mod 143) we see that GCD(7,143) = 1, 7*41 – 2*143 = 287 – 286 = 1
- Thus, the solution is 3*41 = 123 and indeed 123*7 = 3 + 6*143
- Similarly, for 11x = 22 (mod 143) we see that 1*11 +0*143 = 11, and since 22/11 = 2, we see that x = 2 is a solution of this equation

36

Multiple Solutions of ax = b (mod n)

- Suppose ax = b (mod n) and ay = b (mod n), where x and y are < n
- Then a(x-y) = 0 mod n
- So a(x-y) = kn for some k
- We know that g = GCD(a,n) so a = pg and n =qg where GCD(p,q) = 1
- Write pg(x-y) = kqg so p(x-y) = kq, since p and q are coprime, (x-y) = dq and k = dp
- Thus we see that x = y + dq for some d
- If GCD(a,n) = 1, n = q and (x-y) = 0 (mod n), i.e, the solution is unique

37

Multiple Solutions of ax = b (mod n)

- If GCD(a,n), given a solution y we can see that y + q will be another solution
- You can generate all solutions by starting at any solution and adding q = n/g repeatedly
- For 11x = 22, we could start with the solution 2 and repeatedly add 143/11 = 13 to get other solutions
- You can start at any solution and derive all the other solutions by adding n/g repeatedly
- You don't have to start with the smallest root since the repeated additions will just wrap around
- Note that the solutions form a subset of $\mathbb{Z}_n$ that is closed under addition and subtraction, but it need not be a subgroup, since 0 might not be in it

38

Systems of Linear Equations

- Consider the system of linear equations

1. $x = b_1 \pmod{n_1}$
2. $x = b_2 \pmod{n_2}$
3. $x = b_3 \pmod{n_3}$
4. …
5. $x = b_k \pmod{n_k}$

- Note that the coefficients are all 1 so we know each equation has a unique solution (GCD(1,k) = 1 ∀ k)
- When can we find a solution to the entire system of equations
- This leads us to the Chinese Remainder Theorem

39

The Chinese Remainder Theorem – Wikipedia (edited)

- *The Chinese remainder theorem is a theorem of number theory, which states that if one knows the remainders of the division of an integer n by several integers, then one can determine uniquely the remainder of the division of n by the product of these integers, under the condition that the divisors are pairwise coprime.*
- *The earliest known statement of the theorem is by the Chinese mathematician Sunzi in the 3rd century AD (not quite 2000 years).*
- *The Chinese remainder theorem is widely used for computing with large integers, as it allows replacing a computation for which one knows a bound on the size of the result by several similar computations on small integers.*
- *The Chinese remainder theorem (expressed in terms of congruences) is true over every principal ideal domain. It has been generalized to any commutative ring, with a formulation involving ideals.*

40

The Chinese Remainder Theorem

- The original statement was (Wikipedia)
- *There are certain things whose number is unknown. If we count them by threes, we have two left over; by fives, we have three left over; and by sevens, two are left over. How many things are there? (Sunzi, 3rd Century AD)*
- x = 2 (mod 3), x = 3 (mod 5), and x = 2 (mod 7)
- x = 128
- The first complete solution was published by a Chinese mathematician in 1247

41

The Chinese Remainder Theorem (CRT)

- Theorem (CRT): Let s and t be coprimes, then the map $f : \mathbb{Z}_{s*t} \rightarrow \mathbb{Z}_s \times \mathbb{Z}_t$ given by f(m) = (m%s, m%t) is a bijection.
- Proof: We first prove that f is an injection. Suppose f(m) = f(n) for two distinct integers.
- This means that m%s = n%s and m%t = n%t
- This means that (m-n) = 0 (mod s) and (m-n) = 0 (mod t)
- This means that (m-n) is divisible by s and by t
- Since s and t are coprime, (m-n) is divisible by s*t, so m = n (mod s*t)
- Thus, f is an injection
- Note that $|\mathbb{Z}_{s*t}| = s*t = |\mathbb{Z}_s| * |\mathbb{Z}_t| = |\mathbb{Z}_s \times \mathbb{Z}_t|$ so f must be a bijection.

42

Remarks on the CRT

- Note that the proof we gave was non-constructive in the sense that we know there is a unique solution for each set of equations, but we do not know how to find the solution for each set of equations
- We know that if GCD(s,t) = 1, then ∃ a, b such that a*s + b*t = 1
- Note that 1 = 1%s = (a*s+b*t)%s = (b*t)%s and 1 = (a*s)%t
- Suppose we want to solve x%s = m and x%t = n
- Consider the value x = n*a*s + m*b*t
- x%s = (n*a*s+m*b*t)%s = (m*b*t)%s = (m%s)*((b*t)%s) = m%s = m
- Similarly, x%t = n%t = n

43

Remarks on the CRT

- Note that this result is generally not true if GCD(s,t) ≠ 1
- For example, let s = 4 and t = 6 then (1%4,1%6) = (13%4,13%6) and in general (x%4,x%6) = ((x+12)%4,(x+12)%6)
- In general, we want LCM(s,t) = s*t which happens iff s and t are coprime
- Note LCM(s,t) = s*t/GCD(s,t)

44