Algebra Notes

Simon Pratt

Contents

Chapter 1

Relations

1.1 Mappings

A mapping $f : A \to B$ is said to be *onto* or *surjective* if $f(A) = B$.

A mapping (as defined as above) is said to be *one-to-one* or *injective* if $f(a) = f(b) \to a = b$.

A mapping is said to be *bijective* if it is both surjective and injective.

1.2 Equivalence Relation

A relation $R = \{(a, b)|a, b \in A\}$ is said to be *reflexive* if $\forall a \in A : (a, a) \in R$.

A relation (as defined above) is said to be *symmetric* if $(a, b) \in R \to (b, a) \in R$.

A relation is said to be *transitive* if $(a, b) \in Rand(b, c) \in R \to (a, c) \in R$.

A relation is said to be an *equivalence relation* if it is reflexive, symmetric and transitive.

Chapter 2

Division

2.1 The Extended Euclidean Algorithm

Given numbers a, b with which we would like to calculate $gcd(a, b)$, we can do so using the Euclidean Algorithm. Without loss of generality, say $a > b$. Let $r_0 = a, r_1 = b$. We divide r_{i-2}, r_{i-1} to give us $r_{i-2} = r_{i-1} \times q_i + r_i$. We continue dividing until we find the smallest k such that $r_k = 0$, at which point $gcd(a, b) = r_{k-1}$.

The "extension" of the Euclidean Algorithm also gives us s_i, t_i such that $r_i = b \times s_i + a \times t_i$. This is important for *RSA Encryption*. We define $s_0 = 1, s_1 = 0, t_0 = 0, t_1 = 1$ and calculate $s_i = s_{i-2} - q_i s_{i-1}$ and $t_i = t_{i-2} - q_i t_{i-1}$.

We build a table thusly:

i	q_i	r_i	s_i	t_i
0	n/a	39	1	0
1	n/a	14	0	1
2	2	11	1	-2
3	1	3	-3	7
4	3	2	7	-16
5	1	1	-10	23
6	1	0	$\leftarrow$	done

So we know $gcd(14, 39) = 1 = 39 \times -10 + 14 \times 23$.

Chapter 3

Monoids

A *binary operation* is a mapping from $A \times B$ to C.

A binary operation is said to be *closed* if it maps from $A \times A$ to A.

A closed binary operation ($\circ$) is said to be *associative* if $\forall a, b, c \in A : (a \circ b) \circ c = a \circ (b \circ c)$.

A *monoid* is a 3-tuple $(\circ, M, e)$ where $\circ$ is an associative, closed binary operation on M and the identity element $e \in M : \forall m \in M : e \circ m = m$.

An *inverse* is an element $\forall a \in A \exists a' \in A : a \circ a' = e \mathrm{OR} a' \circ a = e$. This element is a *left inverse* if $a' \circ a = e$ or a *right inverse* if $a \circ a' = e$ and is a *full inverse* if it is both a right inverse and a left inverse.

It is easy to prove that a left inverse is unique given right cancelation, a right inverse is unique given left cancelation and a full inverse is unique from either left or right cancelation.

3.1 Row Monomials

We define the *row monomials*, $RM(n)$, as the set of all $0, 1$ matrices with exactly one 1 per row. We will show that $(\circ, RM(n), I_n)$ forms a monoid, where $\circ$ is matrix multiplication, and I_n is the identity for $n \times n$ matrices. Since matrix multiplication is associative in general, we need only prove the following claim:

Claim 3.1. $RM(n)$ is closed under matrix multiplication.

Proof. Let $A, B \in RM(n)$. A is composed of the elements a_{ij}. For all $1 \leq i \leq n$ there exists a unique l such that $a_{il} = 1$ and for all $j \neq l$, $a_{ij} = 0$.

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} = a_{il} b_{lj} = b_{lj}$$

$$\therefore (AB)_{ij} \in RM(n)$$

$\square$

Thus proving $RM(n)$ is a monoid.

3.2 Deterministic Finite Automata

A *Deterministic Finite Automata* (DFA) is a 5-tuple (A, S, s, F, δ) where A is a set of symbols called the *alphabet*, $S = \{s_1, s_2, ..., n\}$ is the set of states, $s \in S$ is the *start state*, $F \subseteq S$ is the set of all accept states, and finally $\delta : A \times S \to S$ is the transition function which returns the next state when given the current state and a symbol to read. A DFA reads a string of characters from the alphabet one character at a time, transitioning from state to state until no more characters can be read. At which point, the DFA either *accepts* the string if the current state is in the set F, or *rejects* the string otherwise.

We can model a DFA using row monomials thusly:

1. to each state $s_k \in S : 1 \leq k \leq n$, we associate a vector such that for $1 \leq i \leq n$: $[s]_i = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases}$
2. to each character $a \in A$, we associate a matrix from $RM(n)$, such that for $1 \leq i \leq n$ and $1 \leq j \leq n$: $[a]_{ij} = \begin{cases} 1 & \text{if } \delta(s_i, a) = s_j \\ 0 & \text{otherwise} \end{cases}$

Chapter 4

Groups

A *group* is a monoid where every element has a full inverse.

The *order* of a group is the number of elements in the group, or infinity if the group is infinite.

A *subgroup* of a group G is a subset of the elements in G that form a group. If H is a subgroup of G, we say $H < G$.

4.1 Cyclic Groups

A *cyclic group* is a group generated by the operation being repeatedly performed on a generator element. In the case of a multiplicative group: $< a >= \{a^k | k \in \mathbb{Z}\}$, in which case a is the generator element.

The order of a cyclic group is the smallest positive integer k such that a^k is the identity.

If $| < a > | = n$, the subgroup $< a^k >=< a^d >$ where $d = gcd(k, n)$ and $| < a^k > | = \frac{n}{gcd(k,n)}$.

4.2 Homomorphisms

A *homomorphism* is a map $\phi : G \to G'$ where $G = (\circ, S, e)$ and $G' = (\otimes, S', e')$ such that $\phi(g) \otimes \phi(h) = \phi(g \circ h)$.

A homomorphism $G \to G$ is an *endomorphism*.

A homomorphism where ϕ is surjective (onto) is called an *epimorphism*.

A homomorphism where ϕ is bijective (onto and one-to-one) is called an *isomorphism*.

Chapter 5

Rubik's Cube Math

5.1 Terminology

First we name the faces of the cube, these are arbitrary but important. We choose a front face F, and an up face U and this uniquely determines the rest of the faces: back B, down D, left L, and right R.

$F =$ white
$U =$ red

Which on my cube gives:

$B =$ yellow
$D =$ orange
$L =$ blue
$R =$ green

We refer to the little cubes that make up the whole cube as "cubies". Each cubie has a unique name determined by its colors. An edge cubie has two colors, and an example of one is the front-up cubie (which in my case is colored white and red), or "fu". A corner cubie has three colors, e.g. fur (white, red, green).

A center cubie only has one color, but since they always stay in the same place relative to other center cubies, we can disregard them except to determine which face is which in an unsolved cube.

Using cycle notation, we can now define operations on the cubes. The move F is to turn the front face clockwise by 90 degrees. Moving the front face anti-clockwise by 90 degrees is the inverse of the F operation, so we consider this F^{-1}. Moving the front face clockwise or anti-clockwise by 180 degrees is either F^2 or F^{-2}, whichever is preferred.

$F =$ (ful fur fdr fdl) (fu fr fd fl)
$U =$ (ful bul bur fur) (fu lu bu ru)
$B =$ (bur bul bdl bdr) (bu bl bd br)
$D =$ (fdl fdr bdr bdl) (fd rd bd ld)
$L =$ (bul ful fdl bdl) (lu fl dl bl)
$R =$ (fur bur bdr fdr) (ru br rd fr)

5.2 Composition

We can define macro operations by compositions of these cyclic functions. For example, the conjugacy of D by F is:

$$FDF^{-1} = \text{(ful fur fdr fdl) (fu fr fd fl) (fdl fdr bdr bdl) (ld bd rd fd) (fdl fdr fur ful) (fl fd fr fu)}$$

But we can consider the edge cubies separately from the corner cubies, since they are disjoint.

$$FDF^{-1} = \text{AB}$$

$$\begin{aligned} A &= \text{(ful fur fdr fdl)(fdl fdr bdr bdl)(fdl fdr fur ful)} \\ &= \text{(ful fdl bdr bdl)(fur)(fdr)} \\ &= \text{(ful fdl bdr bdl)} \end{aligned}$$

$$\begin{aligned} B &= \text{(fu fr fd fl)(ld bd rd fd)(fl fd fr fu)} \\ &= \text{(fl ld bd rd)(fd)(fr)(fu)} \\ &= \text{(fl ld bd rd)} \end{aligned}$$

$$\begin{aligned} FDF^{-1} &= \text{AB} \\ &= \text{(ful fdl bdr bdl)(fl ld bd rd)} \end{aligned}$$

Notice that this move is disjoint to U except in one corner, ful.

Let $M_1 = FDF^{-1}$. We create the commutator of M_1 and U as follows:

$$\begin{aligned} T_1 &= U^{-1}M_1^{-1}UM_1 \\ &= CD \end{aligned}$$

$$\begin{aligned} C &= \text{(fur bur bul ful)(bdl bdr fdl ful)(ful bul bur fur)(ful fdl bdr bdl)} \\ &= \text{(bdl ful fur)(fdl)(bdr)(bul)(bur)} \\ &= \text{(bdl ful fur)} \end{aligned}$$

$$\begin{aligned} D &= \text{(ru bu lu fu)(rd bd ld fl)(fu lu bu ru)(fl ld bd rd)} \\ &= \text{(ru)(bu)(lu)(fu)(rd)(bd)(ld)(fl)} \end{aligned}$$

$$\begin{aligned} T_1 = CD &= \text{(bdl ful fur)(ru)(bu)(lu)(fu)(rd)(bd)(ld)(fl)} \\ &= \text{(bdl ful fur)} \end{aligned}$$

As it turns out, this is the minimal move on corner cubies. Using this, we can create moves on any three corner cubies by conjugation. Let's say we would like to move bul instead of bdl. We can do this using conjugation. First we find a move which moves the cubie we want to move into the cubicle we know how to move. In the example of moving bul of bdl, we need only move bul bdl, so our move Z_1 is simply B. Then we conjugate, ignoring the edge cubies since they will be moved back into position.

$$\begin{aligned} Z_1^{-1}T_1Z_1 &= B^{-1}T_1B \\ &= (\text{bdr bdl bul bur})(\text{bdl ful fur})(\text{bur bul bdl bdr}) \\ &= (\text{bul ful fur}) \end{aligned}$$

So our full move is (read right to left):

$$\begin{aligned} Z_1^{-1}T_1Z_1 &= B^{-1}T_1B \\ &= B^{-1}U^{-1}M_1^{-1}UM_1B \\ &= B^{-1}U^{-1}F^{-1}D^{-1}FUFDF^{-1}B \end{aligned}$$

Now let's say we would like (bdl bur bul). We need to move ful to bur, and fur to bul. This can be done easily with $Z_2 = U^2$, which is conveniently its own inverse.

$$\begin{aligned} U^2 = UU &= (\text{ful bul bur fur})(\text{ful bul bur fur}) \\ &= (\text{ful bur})(\text{bul fur}) \end{aligned}$$

$$\begin{aligned} Z_2^{-1}T_1Z_2 = U^2T_1U^2 &= (\text{ful bur})(\text{bul fur})(\text{bdl ful fur})(\text{ful bur})(\text{bul fur}) \\ &= (\text{bul bdl bur}) \end{aligned}$$

So our full move is (read right to left):

$$\begin{aligned} Z_2^{-1}T_1Z_2 &= U^2T_1U^2 \\ &= U^2U^{-1}M_1^{-1}UM_1U^2 \\ &= UF^{-1}D^{-1}FUFDF^{-1}U^2 \end{aligned}$$