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 $e \in M : \forall m \in M : e \circ m = m$.

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

Rubik's Cube Math

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

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

$$\begin{aligned}F^{-1}DF &= \text{(fdl fdr fur ful) (fl fd fr fu) (fdl fdr bdr bdl) (fd rd bd ld) (ful fur fdr fdl) (fu fr fd fl)}\\ &= \text{(fur bdr bdl fdr) (ful) (fdl) (fr rd bd ld) (fu) (fl)}\\ &= \text{(fur bdr bdl fdr) (fr rd bd ld)}\end{aligned}$$

And the commutator of F and D is:

$$\begin{aligned}F^{-1}D^{-1}FD =&\text{(fdl fdr fur ful) (fl fd fr fu) (bdl bdr fdr fdl) (ld bd rd fd)}\\ &\text{(ful fur fdr fdl) (fu fr fd fl) (fdl fdr bdr bdl) (fd rd bd ld)}\\ =&\text{(fdl bdl) (fdr fur) (bdr) (ful) (fd fr ld) (rd) (bd) (fu) (fl)}\\ =&\text{(fdl bdl) (fdr fur) (fd fr ld)}\end{aligned}$$

Remembering that $(F^{-1}D^{-1}FD)^{-1} = D^{-1}F^{-1}DF$.

And since 2 cycles are their own inverses, we can run this move twice to attain:

$$\begin{aligned}M = F^{-1}D^{-1}FDF^{-1}D^{-1}FD =&\text{(fdl bdl) (fdr fur) (fd fr ld) (fdl bdl) (fdr fur) (fd fr ld)}\\ =&\text{(fd ld fr)}\end{aligned}$$

We now have a minimal move on the edge cubies. For those following along, you may notice that the corner cubies are in the correct location but twisted, which our notation does not capture.