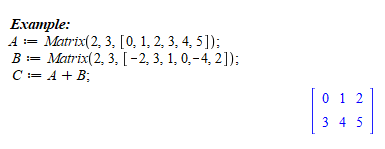
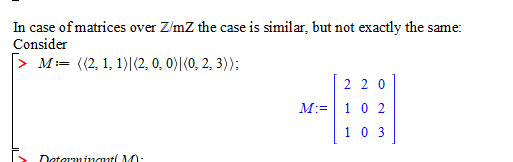
Matrice as Rings ，矩阵作为Ring

首先怎么打一个矩阵



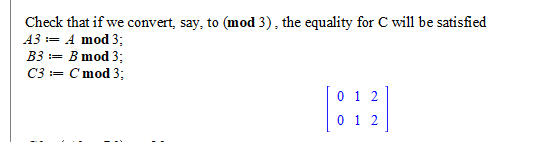
（另外一种打矩阵的方法）

上面这个是A

矩阵里的元素可以是Z整数，，也也可以是R实数，我们甚至能让里面的元素是commutative ring的元素。 //矩阵本身不一定是ring

Z/pZ, 对矩阵每个元素进行mod P

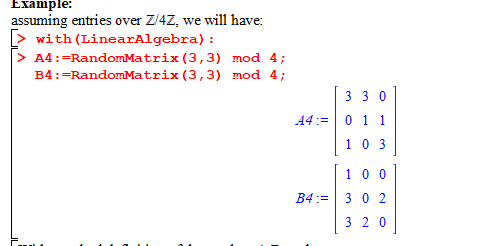
因为矩阵完美满足加法五规则

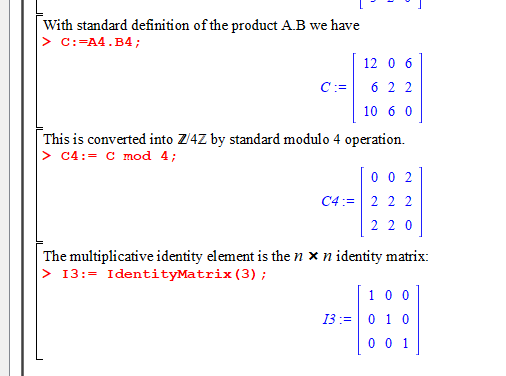
所以Z/pZ会形成一个group  
这个是对A进行·mod 3

因为是group所以在矩阵内



如果想让一个matrice从group进化成ring，必须让他行数列数相等





identityMatrix就是矩阵的乘法identity

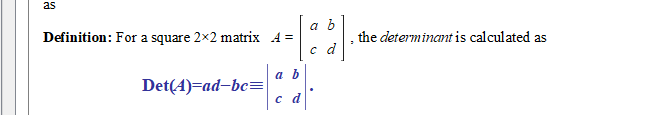
可以满足乘法的结合律因为这个矩阵的每一个元素都是建立在commutative ring上的

同样的，也能满足distributivity

所以这样的一个矩阵是标准的ring，但不是commutative ring,因为AB≠BA

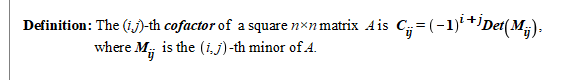
Determinants and inverses

determinant:



cofactor 余子式矩阵

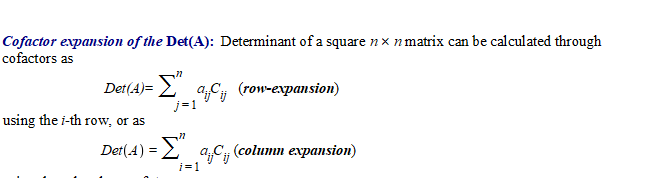
一个matrix的 (ij)-th cofactor 等于



Minor Mij

去除那一行那一列新组成的matrix



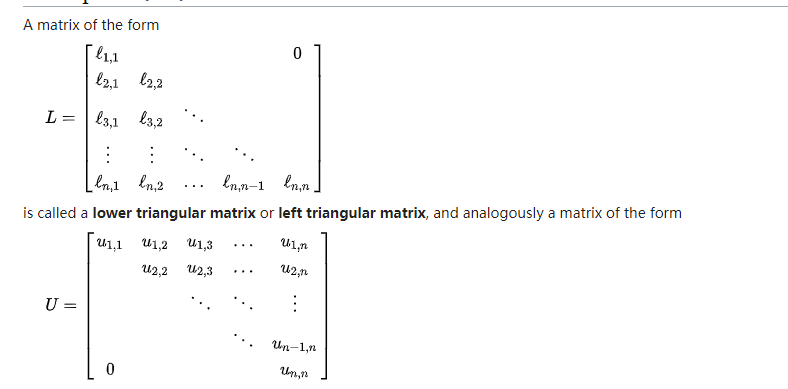


Cofactor expansion of the Det(A)

一个矩阵的值等于不停的切割行或列，系数为被切割的元素aij

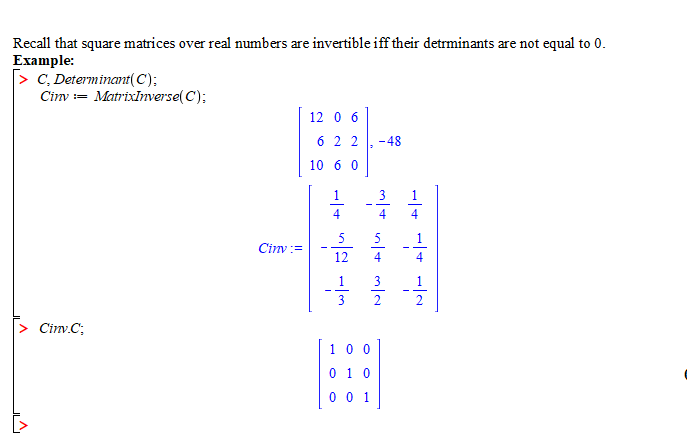
推论

对于任何三角矩阵



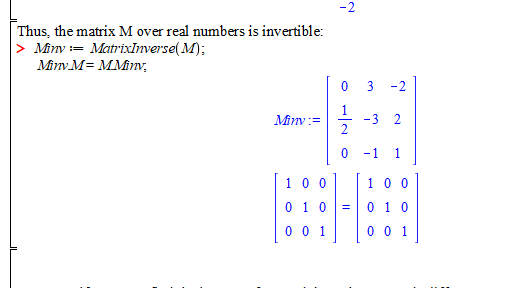
值等于斜着的部分相乘

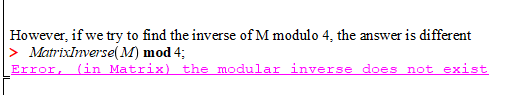




用Determinant就可以直接算出一个矩阵的值

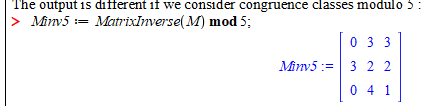
MatrixInverse可以直接算出逆矩阵（逆矩阵乘以原矩阵等于标准矩阵//这个可以倒过来）





Matrixinverse（M） mod5 有解，mod4 无解

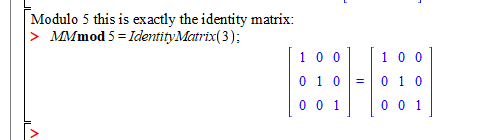
定义modular inverse of matrix



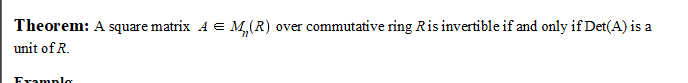
在后面加上mod5以后生成的东西与原来的矩阵相乘



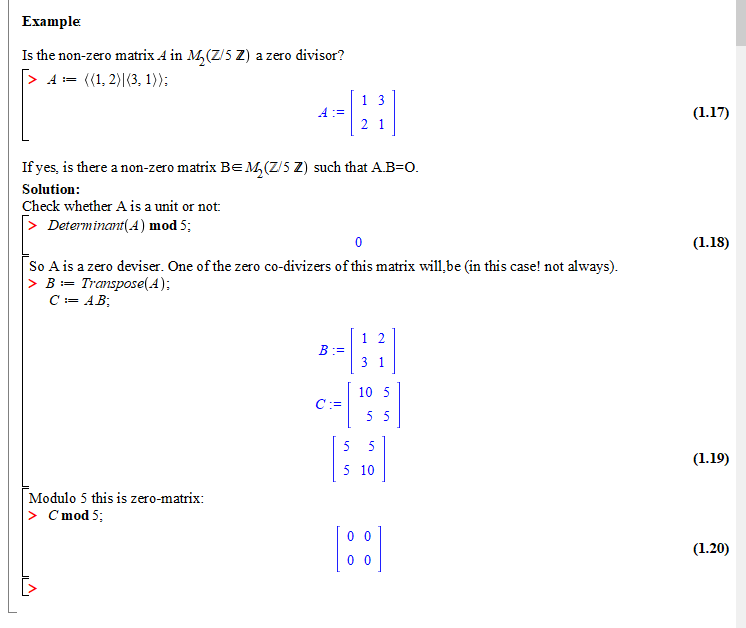
得到的新矩阵mod 5将得到标准矩阵



为什么Mod 4不行mod5可以



如果Det(A) 为R的uinit，那么这个矩阵是Invertible的



怎么确认一个matrix是不是zero divisor

Det(A) mod 5=0

CH8

应用1:error-correcting code

一个信息通常以binary block二进制block在一个嘈杂的频道中传递，有可能产生error

那么以下可能吗：

（1）检测到有error的binary block

（2）纠正这个error

Parity check code:

假设Sandra想传递一个n位 block of data abcde

他会传递abcdef//总之就是在原有基础上多加一位，f=a+b+c+d+...e(mod 2) 额外的digit f 检查了原信息的奇偶parity 如果有一个error也就是一位数改变了，那么parity就改变了， 但是这只能检测到一个error也不能纠正error,也不知道具体位置

Repetition code

重复代码

每一个信息digit都有充足的redundancy备用， 例如我们发出digit a，就发出aaaaa， 这样如果有error就能检测并改正，如果被传递的data小于等于(n-1)/2//五位错了两位能纠正，三位就挣不了， 如果收到了aaabb，那么结论就是a，但是效率很低为1/n

Humming error-detection and correction codes

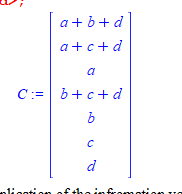
Code 1.



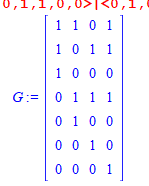
如果Sandra想传递一个r=4的digit abcd

它构建了一个vector，在原有的4位又加了3位，（parity digits），xyz  

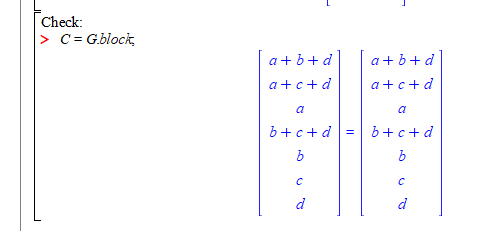

那么他们的vector可以写成



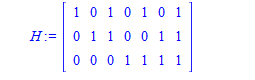
我们甚至可以通过abcd位形成information block G



G.block指令会把一行的数加在一起



当Peter 收到了vector R, 然后他需要应用另一个矩阵H，来检查R是不是原始矩阵C



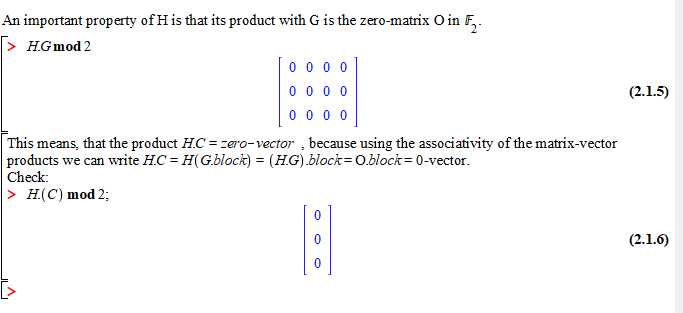
这个H怎么生成的，H从下往上三排对应G从左往右三列，然后1与0倒转

这个H的读法是一列一列column读，从下往上读，例如第六列就代表110代表6

H的重要性质：

H\*G mod 2 为 zeromatrix

H\*C mod 2 为zero vector

  
所以如果收到正确的vector，那么相乘应该是0向量，

然而如果有一个error，既收到的向量不是原向量，那么乘积就不会0

假设收到的R与原来的C不同，

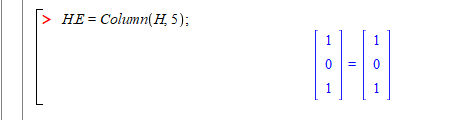
那么我们引入difference vector E

//为什么要Mod 2, 因为 不同有可能是0，1 这种情况，Mod2以后都是1

我们就知道是第五DIGIT出了问题

HR=H(E+C)=HE=0=HE

而HE实际就等于H的第五列

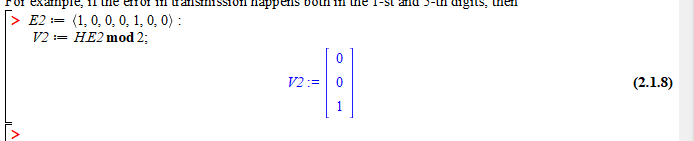


然后我们就能意识到error在位置

这个代码很有效，但是如果多于一个error，就会失效

例如

如果第一个digit与第五个digit同时为error

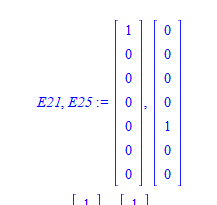


E2就变成对应的1，0,0,0,1,0,0

那么这时HE2 mod2 =[0,0,1]，算出来位置为4

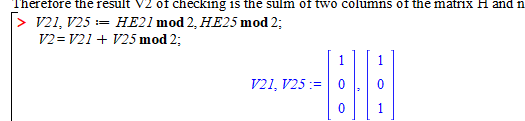
Misleading

所以我们应该吧E2拆开来

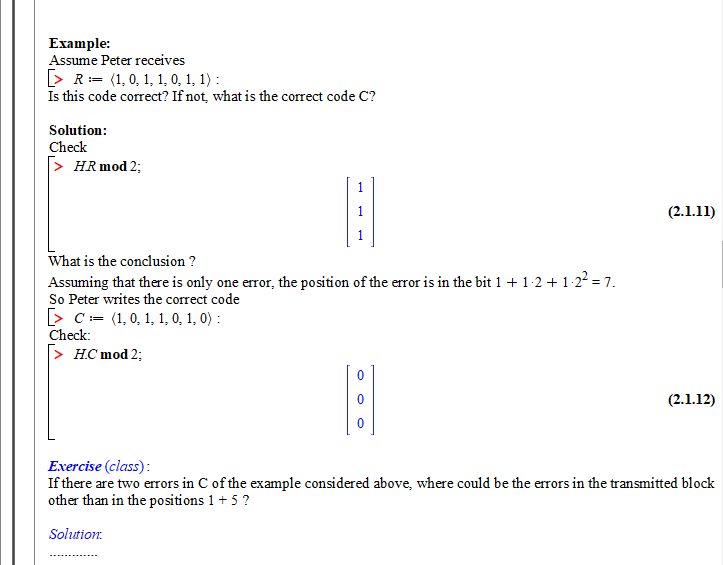


一个vector只能有一个1

然后再分别乘



位置分别是1， 5

例题：

根据收到的R，我们直接让HR mod 2 //H是原来给的H

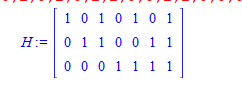
得到了一个[1 1 1]

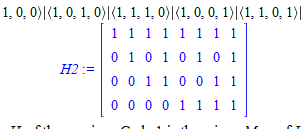
假设只有一个ERROR， 那么位置就是7

那么我们就要改成1，0，1，1，0，1，1

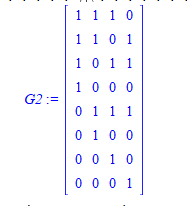
.

Code2 ，如果有多个error，那么code 1实际上是会fail的，一个基于员阿里版本的code2允许检测到2个error。然后纠正一个error。

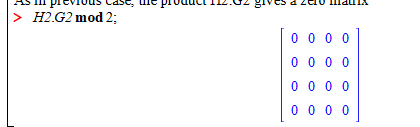


在原来基础上新加一行一列，第一行1，左边列0

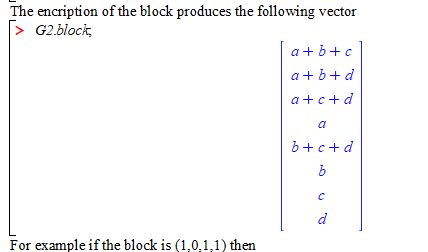
G与H对应，G从左往右，H从下往上 ，1变0，0变1



这样H与G仍然能生成zero matrix



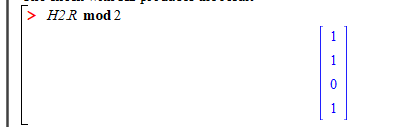
G2.block合并



真实C为00110011

这时假设收到的R为，，错误在第六位

检查错位位置

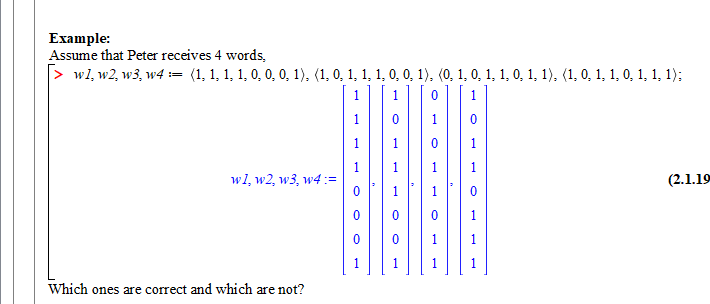


1+1\*2^0+0\*2+1\*2^2=6

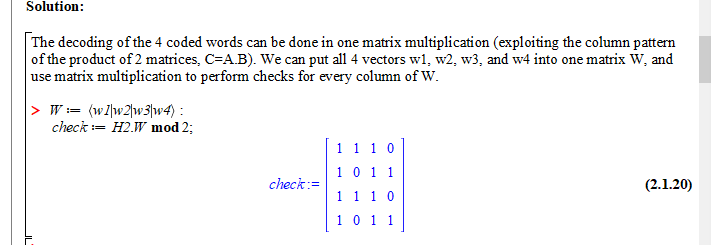
如果有两个error，那么这个vector的第一个元素将是0，但是纠正是不可能的

例如

Peter收到了以下4个word



检查有没有error的方法永远是让H\*w

因为没有一个0vector所以都是error

w1w2w3都是只有一个error，直接计算位置

1+1\*1+1\*2+2\*2=8

1+1\*2=3

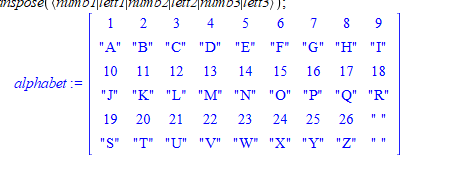
1+1\*1+1\*2+2\*2=8

0开头完全无法check

Application 2

Hill Cryptosystem

用一个基于ring的矩阵来表示字母表



假设Sandra想要发送信息 Buy ENRON, 那么会把这个信息转换成一串数字P， 然后再把P利用某个function加密成另一串乱码C 给Peter。

Encrypt:加密,利用function C=f(p)

inverse function 只有Peter能知道，为了保密。

Decrypt:解密 这样Peter就能通过他的inverse，利用收到的C（乱码）与inverse function得到P，

首先我们要选好n阶加密矩阵A

利用(A\*P )mod26来计算出加密后的C

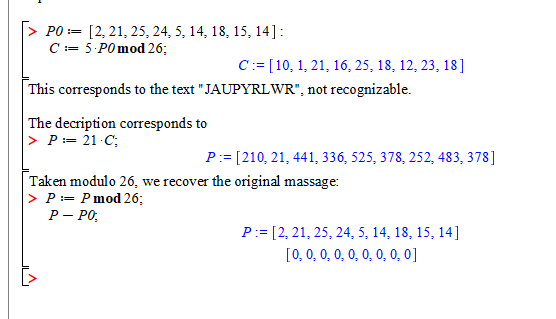
然后再利用(A的逆矩阵乘C)mod 26得到P //不一定有逆矩阵，

P有时候不能直接使用当n不是1

要把P截成长度为n的vector

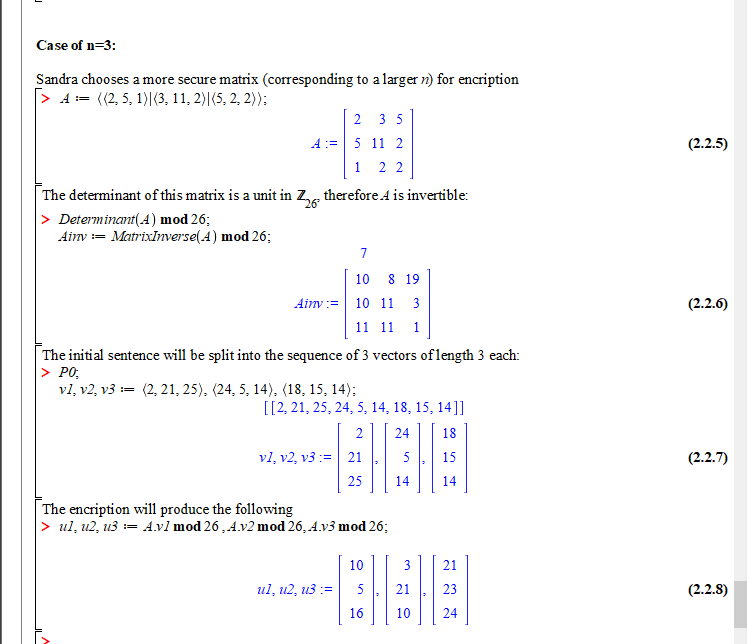
例如这里，这句话变成了对应的数字=

当n=1,A=[5]



加密成的C是乱码，【5】的逆矩阵是【21】

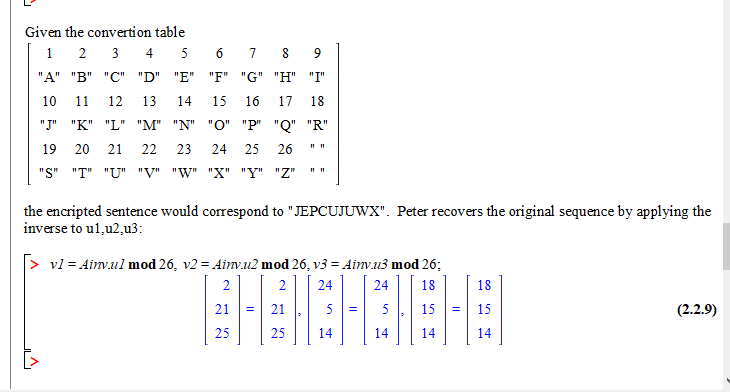
[21]\*C得到P，在mod26就是原信息



例如n=3，我们先要用determinant(A) 是不是Z26的unit， 如果是，才是invertible，才有意义

然后把原式子切割成对应的大小这样才能让A乘他

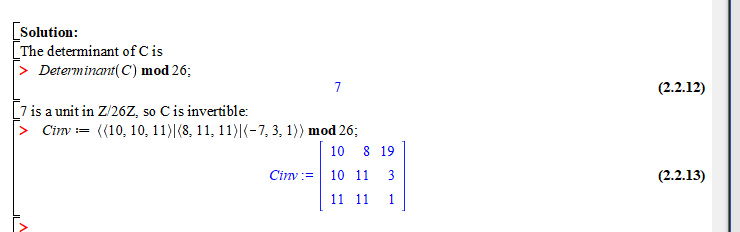
u1u2u3就是我们的初始信息



然后再乘回去转化出对应的信息

例题：给了你转换matrix，让你解码一个数

首先看能不能inv



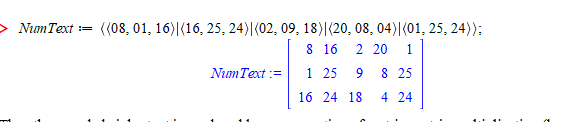
然后将matrix转换成对应的数字



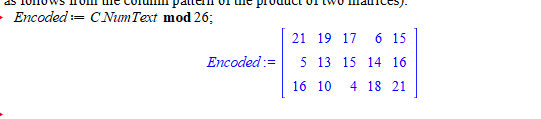
一个数字占两位，08代表着8

因为是3x3

所以我们需要切割text



加密Encode就是Cx信息



解码就是inverse乘以加密信息

