Appendices: Using noncommutative Gröbner bases in solving partially prescribed matrix inverse completion problems

August 31, 1999

We intend these appendices to appear on the internet for those interested and not be included in the publication.

Abstract

We investigate the use of noncommutative Gröbner computer algebra in solving partially prescribed matrix inverse completion problems. The type of problems considered here are similar to those in [BLJW]. There the authors, among other things, gave necessary and sufficient conditions for the solution of a two by two block matrix completion problem. Our approach is exceedingly different than theirs and relies on symbolic computer algebra.

Here we describe a general method by which all block matrix completion problems of this type may be analyzed with sufficient computational power. We also demonstrate our method with an analysis of all three by three block matrix inverse completion problems with eleven known blocks and seven unknown. We discover that the solutions to all such problems are of a relatively simple form.

We then do a more detailed analysis of what can be considered the "central problem" of the 31,824 three by three block matrix completion problems with eleven known blocks and seven unknown. A solution to this problem of the form derived in [BLJW] is presented.

Not only do we give a proof of our detailed result, but we describe the strategy used in discovering our theorem and proof since it is somewhat unusual for these types of problems.

1 Appendix 1 - Mathematica implementation of pseudo code given for Theorem 1

```
(* First we set up needed variables and lists *)
onePerms = Permutations[{0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1}];

vars = { k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,u1,u2,u3,u4,u5,u6,u7 };

SetNonCommutative[ vars ];

SetNonCommutative[ Inv[k1],Inv[k2],Inv[k3],Inv[k4],Inv[k5],Inv[k6],Inv[k7],
Inv[k8],Inv[k9],Inv[k10],Inv[k11] ];

inversesKeep =
{-1+k1**Inv[k1],-1+k10**Inv[k10],-1+k11**Inv[k11],-1+k2**Inv[k2],
-1+k3**Inv[k3],-1+k4**Inv[k4],-1+k5**Inv[k5],-1+k6**Inv[k6],
-1+k7**Inv[k7],-1+k8**Inv[k8],-1+k9**Inv[k9],-1+Inv[k1]**k1,
-1+Inv[k10]**k10,-1+Inv[k11]**k11,-1+Inv[k2]**k2,-1+Inv[k3]**k3,
-1+Inv[k4]**k4,-1+Inv[k5]**k5,-1+Inv[k6]**k6,-1+Inv[k7]**k7,
-1+Inv[k8]**k8,-1+Inv[k9]**k9 };

permMtcs = Permutations[ IdentityMatrix[3] ];

invPermMtcs = Map[ Inverse, permMtcs ];

knownVars = {k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11};
unkVars = {u1,u2,u3,u4,u5,u6,u7 };

alreadyDone = {} ;


(* Here we give some necessary functions *)

(* Make symbolic matrix from 1's notation *)
MakeSymbMatrix[ onesList_List ] := Module[{idx,unkIdx=1,knIdx=1,newMtx={} },

  For[idx=1,idx<=Length[onesList],idx++,

    If[ onesList[[idx]] == 1,
      AppendTo[ newMtx,unkVars[[unkIdx]] ];
      unkIdx++;,
      AppendTo[ newMtx,knownVars[[knIdx]] ];
      knIdx++;
      ];
    ];
  Return[ newMtx ];
];


(* Ask if a permutation of a matrix is in the alreadyDone list *)
PermMemberQ[  onesList_List ] :=
Module[ {permList={}, A,B},
   A = Partition[onesList,9 ][[1]];
   B = Partition[onesList,9 ][[2]];
   A = Partition[A,3];
   B = Partition[B,3];

   Apermed = Map[ A.#  & ,permMtcs] ;

   Aperm2 = Flatten[Table[Map[#.Apermed[[permEntry]]&, permMtcs],
{permEntry,1, Length[Apermed]}],1];
```

```
   Bpermed = Map[ #.B & , invPermMtcs ];
   Bperm2 = Flatten[ Table[Map[ Bpermed[[permEntry]].# &, invPermMtcs ],
{ permEntry,1,Length[Bpermed]}],1];
   For[index =1,index<=Length[Aperm2],index++,
   AppendTo[permList, Flatten[{Aperm2[[index]],Bperm2[[index]]}]];
   ];

   If[ Intersection[ alreadyDone,permList ] === {},
     Return[ False ];,
     Return[ True  ];
    ];
];

(* Find the unknown indeterminates which lie in equations in one unknown *)
FindDetermined[ aList_List,currentUnknowns_List ,opts___Rule] :=
Module[{i,len,item,vars,kn,unk,n,alllengths,rules, determList,
        allvars,allind,j,totalvars,outputToFile,fileName},
   Clear[relations];
    determList = {};

   rules = Union[ExpandNonCommutativeMultiply[aList]];
   rules = PolyToRule[rules];
   rules = Union[rules];
   len   = Length[rules];
Do[item = rules[[i]];

     If[Not[item===0],
         vars = GrabIndeterminants[item];

          If[ Length[ Union[vars] ] == 1,
               AppendTo[determList , vars[[1]] ];
          ];
      ];
   ,{i,1,len}];  (* End Do[] loop *)
   Return[ determList ];
];

(* Find the indeterminates which are singletons *)
FindSingletons[ aList_List,currentUnknowns_List ,opts___Rule] :=
Module[{i,len,item,vars,kn,unk,n,alllengths,rules, singletonList,
        allvars,allind,j,totalvars,outputToFile,fileName},
   Clear[relations];
   singletonList = {};
   rules = Union[ExpandNonCommutativeMultiply[aList]];
   rules = PolyToRule[rules];
   rules = Union[rules];
   len   = Length[rules];
Do[item = rules[[i]];

     If[Not[item===0],
        vars = GrabIndeterminants[item];

         (* If the head is not NCM it's a singleton !! *)
          If[Head[item[[1]]]=!=NonCommutativeMultiply,
```

```
AppendTo[singletonList, item[[1]] ];
          ];
      ];
   ,{i,1,len}];  (* End Do[] loop *)
   Return[Intersection[ singletonList, currentUnknowns ]
   ];
];

<<Extra.TeXForm;
WriteMatrixTeX[ A_List, B_List, currNum_ ] := Module[{},

OpenWriteForTeX["thms/mtcs"<>ToString[currNum] ];
ExpressionToTeXFile["thms/mtcs"<>ToString[currNum]<>".tex" ,
OutputAMatrix[A] ];
ExpressionToTeXFile["thms/mtcs"<>ToString[currNum]<>".tex" ,
OutputAMatrix[B] ];
WriteString["thms/mtcs"<>ToString[currNum]<>".tex",
"\\end{document}" ];

Close["thms/mtcs"<>ToString[currNum]<>".tex"];
];

OpenWrite["AnswersForCompleteMtx"];
undetermList =
 {683,684,689,695,706,708,710,719,725,729,748,749,752,755,760,769,779,784,1656,
  1675,1701,2580,2599,2625,4296,4315,4341,5957,5958,5963,5969,5980,5982,5984,
  5993,5999,6003,6022,6023,6026,6029,6034,6043,6053,6058,6090,6109,6135,6174,
  6193,6219,6294,6313,6339,7299,7318,7344,7486,7487,7491,7498,7499,7500,7501,
  7520,7521,7540,7556,7575,7606,7607,7608,7609,7610,7611,7738,7757,7821,7948,
  7967,8031,9070,9089,9153,10357,10376,10440,11244,11263,11289,12304,12323,
  12349,13409,13410,13414,13421,13422,13423,13424,13443,13444,13463,13479,
  13498,13529,13530,13531,13532,13533,13534,13535,13554,13618,13829,13848,
  13912,14159,14178,14242,15941,15960,16024,16249,16268,16294,17364,17383,
  17447,17943,17962,18026,20312,20331,20357,22914,22915,22919,22926,22927,
  22928,22929,22948,22949,22968,22984,23003,23034,23035,23036,23037,23038,
  23039,23040,23059,23123,23124,23143,23207,23574,23593,23657,24069,24088,
  24152,24257,24276,24302,25372,25391,25455,26786,26805,26869,28954,28973,
  29037,29789,29808,29872,30476,30477,30482,30488,30499,30501,30503,30512,
  30518,30522,30541,30542,30545,30548,30553,30562,30572,30577,30609,30628,
  30654,30693,30712,30738,30813,30832,30858,30978,30997,31023,31198,31217,
  31243,31484,31503,31529} ;


(* Here is the MAIN LOOP *)
For[ currIndex = 1 , currIndex <= Length[onePerms],
currIndex ++,

Write["AnswersForCompleteMtx", "permutation#",
      currIndex ];

If[ PermMemberQ[ onePerms[[currIndex]] ],
    Print[ "Found One Already Done !!!! " ];
    Write["AnswersForCompleteMtx",
      " Found one already done",
      onePerms[[currIndex]] ];

Continue[];
```

```
];

(*  Only add to alreadyDone list if a perm is not in it *)
AppendTo[ alreadyDone,
       onePerms[[currIndex]] ];

If[ MemberQ[undetermList, currIndex  ],
Write["AnswersForCompleteMtx",
          "This problem is of UNDETERMINED FORM. ",
          onePerms[[currIndex]] ];
Continue[];
];

    alreadyDone = Union[ alreadyDone ];
newMtcs = MakeSymbMatrix[onePerms[[currIndex]] ];
mtrcs = Partition[ newMtcs ,9 ];

matrixA = Partition[ mtrcs[[1]] ,3 ];
matrixB = Partition[ mtrcs[[2]] ,3 ];

WriteMatrixTeX[matrixA, matrixB, currIndex ];

oneway = MatMult[matrixA,matrixB] - IdentityMatrix[3];
otherway = MatMult[matrixB,matrixA] - IdentityMatrix[3];
start = Flatten[{ oneway, otherway  }];
inverses = inversesKeep;
start=Join[start,inverses];

ClearMonomialOrderAll[];
  SetMonomialOrder[ {{k1,Inv[k1],k2,Inv[k2],k3,Inv[k3],k4,Inv[k4],
k5,Inv[k5],k6,Inv[k6],k7,Inv[k7],k8,Inv[k8],k9,Inv[k9],
k10,Inv[k10],k11,Inv[k11]},{u1},{u2},{u3},{u4},{u5},{u6},{u7}}];

fileName = "thms/WoerdOutput-"<>ToString[currIndex];
    (* This function creates a Noncommutative Groebner basis and
a TeX file describing the output *)
answer = NCProcess[start,2,2,1,1,
      fileName ,RR->True, SB->True, SBByCat->False, NCCV->False
] ;

If[  Complement[unkVars, Union[
  FindDetermined[answer[[3]], unkVars],
FindSingletons[ answer[[3]], unkVars] ] ] =!= {} ,
Write["AnswersForCompleteMtx", "Didn't WORK correctly.",
    onePerms[[currIndex]]    ];,
(* else *)

Write["AnswersForCompleteMtx", "Worked fine.",
    onePerms[[currIndex]]    ];

     ];   (*  end If[]  *)

Open["thms/outRels"<>ToString[currIndex]<>".m"];
Write["thms/outRels"<>ToString[currIndex]<>".m",
  answer[[3]] ];
```

```
Close["thms/outRels"<>ToString[currIndex]<>".m"];
]; (* end for *)

Close[ "AnswersForCompleteMtx"];
```

2 Appendix 2 - The First Run in the Discovery Process

Input =
$-1 + a\,x + b\,j + t\,h$
$a\,f + b\,k + t\,y$
$a\,g + b\,z + t\,i$
$c\,h + u\,x + v\,j$
$-1 + c\,y + u\,f + v\,k$
$c\,i + u\,g + v\,z$
$d\,x + e\,j + w\,h$
$d\,f + e\,k + w\,y$
$-1 + d\,g + e\,z + w\,i$
$-1 + f\,u + g\,d + x\,a$
$f\,c + g\,w + x\,t$
$f\,v + g\,e + x\,b$
$h\,a + i\,d + y\,u$
$-1 + h\,t + i\,w + y\,c$
$h\,b + i\,e + y\,v$
$j\,a + k\,u + z\,d$
$j\,t + k\,c + z\,w$
$-1 + j\,b + k\,v + z\,e$
$a\,a^{-1} == 1$
$a^{-1}\,a == 1$
$b\,b^{-1} == 1$
$b^{-1}\,b == 1$
$c\,c^{-1} == 1$
$c^{-1}\,c == 1$
$d\,d^{-1} == 1$
$d^{-1}\,d == 1$
$e\,e^{-1} == 1$
$e^{-1}\,e == 1$
$f\,f^{-1} == 1$
$f^{-1}\,f == 1$
$g\,g^{-1} == 1$
$g^{-1}\,g == 1$
$h\,h^{-1} == 1$
$h^{-1}\,h == 1$
$i\,i^{-1} == 1$
$i^{-1}\,i == 1$
$j\,j^{-1} == 1$
$j^{-1}\,j == 1$
$k\,k^{-1} == 1$
$k^{-1}\,k == 1$
File Name = MatrixInverseAnswer-3
NCMakeGB Iterations = 2
NCMakeGB on Digested Iterations = 3
SmallBasis Iterations = 3
SmallBasis on Knowns Iterations = 4
Deselect→ {}
UserSelect→ {}

RR→True
RRByCat→False
SB→False
SBByCat→False
DegreeCap→12
DegreeSumCap→80
DegreeCapSB→13
DegreeSumCapSB→81
NCCV→False
THE ORDER IS NOW THE FOLLOWING:
$a < a^{-1} < b < b^{-1} < c < c^{-1} < d < d^{-1} < e < e^{-1} < f < f^{-1} < g < g^{-1} < h < h^{-1} < i < i^{-1} < j < j^{-1} < k < k^{-1} \ll z \ll y < x < t < u < v < w$

YOUR SESSION HAS DIGESTED
THE FOLLOWING RELATIONS

THE FOLLOWING VARIABLES HAVE BEEN SOLVED FOR:
$\{t, u, v, w, x, y\}$

The corresponding rules are the following:

$t \to -1\, a\, g\, i^{-1} - b\, z\, i^{-1}$

$u \to -1\, k^{-1}\, j\, a - k^{-1}\, z\, d$

$v \to k^{-1} - k^{-1}\, j\, b - k^{-1}\, z\, e$

$w \to i^{-1} - d\, g\, i^{-1} - e\, z\, i^{-1}$

$x \to a^{-1} + f\, k^{-1}\, j - g\, d\, a^{-1} + f\, k^{-1}\, z\, d\, a^{-1}$

$y \to c^{-1}\, k^{-1}\, j\, a\, f + c^{-1}\, k^{-1}\, j\, b\, k + c^{-1}\, k^{-1}\, z\, d\, f + c^{-1}\, k^{-1}\, z\, e\, k$

The expressions with unknown variables {}
and knowns $\{a, b, c, d, e, f, g, h, i, j, k, a^{-1}, b^{-1}, c^{-1}, d^{-1}, e^{-1}, f^{-1}, g^{-1}, h^{-1}, i^{-1}, j^{-1}, k^{-1}\}$

$a\, a^{-1} \to 1$

$b\, b^{-1} \to 1$

$c\, c^{-1} \to 1$

$d\, d^{-1} \to 1$

$e\, e^{-1} \to 1$

$f\, f^{-1} \to 1$

$g\, g^{-1} \to 1$

$h\, h^{-1} \to 1$

$i\, i^{-1} \to 1$

$j\, j^{-1} \to 1$

$k\, k^{-1} \to 1$

$a^{-1}\, a \to 1$

$b^{-1}\, b \to 1$

$c^{-1}\, c \to 1$

$d^{-1}\, d \to 1$

$e^{-1}\, e \to 1$

$f^{-1}\, f \to 1$

$g^{-1}\,g \to 1$

$h^{-1}\,h \to 1$

$i^{-1}\,i \to 1$

$j^{-1}\,j \to 1$

$k^{-1}\,k \to 1$

USER CREATIONS APPEAR BELOW

SOME RELATIONS WHICH APPEAR BELOW MAY BE UNDIGESTED

THE FOLLOWING VARIABLES HAVE NOT BEEN SOLVED FOR:
$\{a, b, c, d, e, f, g, h, i, j, k, z, a^{-1}, b^{-1}, c^{-1}, d^{-1}, e^{-1}, f^{-1}, g^{-1}, h^{-1}, i^{-1}, j^{-1}, k^{-1}\}$

The expressions with unknown variables $\{z\}$
and knowns $\{a, b, c, d, e, f, g, h, i, j, k, a^{-1}, b^{-1}, c^{-1}, d^{-1}, e^{-1}, f^{-1}, h^{-1}, i^{-1}, k^{-1}\}$

$z\,e\,z \to z - j\,a\,g - j\,b\,z + k\,c\,i - z\,d\,g$

$d^{-1}\,e\,z\,i^{-1} \to a^{-1}\,h^{-1} + d^{-1}\,i^{-1} - a^{-1}\,b\,j\,h^{-1} + a^{-1}\,b\,z\,i^{-1} + d^{-1}\,e\,j\,h^{-1}$

$k^{-1}\,z\,e\,b^{-1} \to f^{-1}\,a^{-1} + k^{-1}\,b^{-1} - f^{-1}\,g\,d\,a^{-1} + f^{-1}\,g\,e\,b^{-1} + k^{-1}\,z\,d\,a^{-1}$

$i\,e\,z\,i^{-1}\,c^{-1} \to -1\,h\,a\,g\,i^{-1}\,c^{-1} - h\,b\,z\,i^{-1}\,c^{-1} - i\,d\,g\,i^{-1}\,c^{-1} + c^{-1}\,k^{-1}\,j\,a\,f + c^{-1}\,k^{-1}\,j\,b\,k + c^{-1}\,k^{-1}\,z\,d\,f + c^{-1}\,k^{-1}\,z\,e\,k$

$k^{-1}\,z\,d\,a^{-1}\,b \to -1\,k^{-1} - f^{-1}\,g\,e - f^{-1}\,a^{-1}\,b + k^{-1}\,z\,e + f^{-1}\,g\,d\,a^{-1}\,b$

$k^{-1}\,z\,e\,k\,c \to c\,h\,a\,g\,i^{-1} + c\,h\,b\,z\,i^{-1} + c\,i\,d\,g\,i^{-1} + c\,i\,e\,z\,i^{-1} - k^{-1}\,j\,a\,f\,c - k^{-1}\,j\,b\,k\,c - k^{-1}\,z\,d\,f\,c$

$a\,f\,k^{-1}\,z\,d\,a^{-1} \to -1\,b\,j - a\,f\,k^{-1}\,j + a\,g\,d\,a^{-1} + a\,g\,i^{-1}\,h + b\,z\,i^{-1}\,h$

$d\,f\,k^{-1}\,z\,d\,a^{-1} \to -1\,d\,a^{-1} - e\,j - i^{-1}\,h - d\,f\,k^{-1}\,j + d\,g\,d\,a^{-1} + d\,g\,i^{-1}\,h + e\,z\,i^{-1}\,h$

$f\,k^{-1}\,z\,d\,a^{-1}\,h^{-1} \to g\,i^{-1} - f\,k^{-1}\,j\,h^{-1} + g\,d\,a^{-1}\,h^{-1} - a^{-1}\,b\,j\,h^{-1} + a^{-1}\,b\,z\,i^{-1}$

$i\,e\,z\,i^{-1}\,h\,a \to h\,a + i\,d - h\,a\,g\,d + h\,b\,j\,a - i\,d\,g\,d + i\,e\,j\,a + h\,a\,f\,k^{-1}\,j\,a + h\,a\,f\,k^{-1}\,z\,d - h\,a\,g\,i^{-1}\,h\,a - h\,b\,z\,i^{-1}\,h\,a + i\,d\,f\,k^{-1}\,j\,a + i\,d\,f\,k^{-1}\,z\,d - i\,d\,g\,i^{-1}\,h\,a$

$i\,e\,z\,i^{-1}\,h\,b \to -1\,h\,a\,f\,k^{-1} - h\,a\,g\,e + h\,b\,j\,b - i\,d\,f\,k^{-1} - i\,d\,g\,e + i\,e\,j\,b + h\,a\,f\,k^{-1}\,j\,b + h\,a\,f\,k^{-1}\,z\,e - h\,a\,g\,i^{-1}\,h\,b - h\,b\,z\,i^{-1}\,h\,b + i\,d\,f\,k^{-1}\,j\,b + i\,d\,f\,k^{-1}\,z\,e - i\,d\,g\,i^{-1}\,h\,b$

$k\,c\,i\,e\,z\,i^{-1} \to j\,a\,f\,c + j\,b\,k\,c + z\,d\,f\,c + z\,e\,k\,c - k\,c\,h\,a\,g\,i^{-1} - k\,c\,h\,b\,z\,i^{-1} - k\,c\,i\,d\,g\,i^{-1}$

$z\,d\,f\,k^{-1}\,z\,d \to -1\,z\,d + j\,a\,g\,d - j\,b\,j\,a + k\,c\,h\,a + z\,d\,g\,d - z\,e\,j\,a - j\,a\,f\,k^{-1}\,j\,a - j\,a\,f\,k^{-1}\,z\,d - z\,d\,f\,k^{-1}\,j\,a$

$z\,d\,f\,k^{-1}\,z\,e \to j\,b + j\,a\,f\,k^{-1} + j\,a\,g\,e - j\,b\,j\,b + k\,c\,h\,b + z\,d\,f\,k^{-1} + z\,d\,g\,e - z\,e\,j\,b - j\,a\,f\,k^{-1}\,j\,b - j\,a\,f\,k^{-1}\,z\,e - z\,d\,f\,k^{-1}\,j\,b$

$k^{-1}\,z\,d\,f\,k^{-1}\,z \to -1\,k^{-1}\,z + c\,h\,a\,d^{-1} + k^{-1}\,j\,a\,g + k^{-1}\,z\,d\,g - k^{-1}\,j\,a\,f\,k^{-1}\,z - k^{-1}\,j\,b\,j\,a\,d^{-1} - k^{-1}\,z\,e\,j\,a\,d^{-1} - k^{-1}\,j\,a\,f\,k^{-1}\,j\,a\,d^{-1} - k^{-1}\,z\,d\,f\,k^{-1}\,j\,a\,d^{-1}$

$b\,z\,i^{-1}\,h\,b\,z\,i^{-1} \to a\,f\,c - b\,z\,i^{-1} - a\,g\,d\,g\,i^{-1} - a\,g\,e\,z\,i^{-1} + b\,j\,a\,g\,i^{-1} + b\,j\,b\,z\,i^{-1} - a\,g\,i^{-1}\,h\,a\,g\,i^{-1} - a\,g\,i^{-1}\,h\,b\,z\,i^{-1} - b\,z\,i^{-1}\,h\,a\,g\,i^{-1}$

$e\,z\,i^{-1}\,h\,b\,z\,i^{-1} \to d\,f\,c + d\,g\,i^{-1} - d\,g\,d\,g\,i^{-1} - d\,g\,e\,z\,i^{-1} + e\,j\,a\,g\,i^{-1} + e\,j\,b\,z\,i^{-1} + i^{-1}\,h\,a\,g\,i^{-1} + i^{-1}\,h\,b\,z\,i^{-1} - d\,g\,i^{-1}\,h\,a\,g\,i^{-1} - d\,g\,i^{-1}\,h\,b\,z\,i^{-1} - e\,z\,i^{-1}\,h\,a\,g\,i^{-1}$

$z\,i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1} \to -1\,z\,i^{-1}\,c^{-1} + b^{-1}\,a\,f + j\,a\,g\,i^{-1}\,c^{-1} + j\,b\,z\,i^{-1}\,c^{-1} - z\,i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1} - b^{-1}\,a\,g\,d\,g\,i^{-1}\,c^{-1} - b^{-1}\,a\,g\,e\,z\,i^{-1}\,c^{-1} - b^{-1}\,a\,g\,i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1} - b^{-1}\,a\,g\,i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1}$

$k^{-1}\,z\,d\,f\,k^{-1}\,j\,b\,e^{-1} \to k^{-1}\,z - c\,h\,a\,d^{-1} + c\,h\,b\,e^{-1} + k^{-1}\,j\,b\,e^{-1} + k^{-1}\,j\,a\,f\,k^{-1}\,e^{-1} + k^{-1}\,j\,b\,j\,a\,d^{-1} - k^{-1}\,j\,b\,j\,b\,e^{-1} + k^{-1}\,z\,d\,f\,k^{-1}\,e^{-1} + k^{-1}\,z\,e\,j\,a\,d^{-1} - k^{-1}\,z\,e\,j\,b\,e^{-1} + k^{-1}\,j\,a\,f\,k^{-1}\,j\,a\,d^{-1} - k^{-1}\,j\,a\,f\,k^{-1}\,j\,b\,e^{-1} + k^{-1}\,z\,d\,f\,k^{-1}\,j\,a\,d^{-1}$

$e^{-1}\,d\,g\,i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1} \;\to\; z\,i^{-1}\,c^{-1} - b^{-1}\,a\,f + e^{-1}\,d\,f + e^{-1}\,d\,g\,i^{-1}\,c^{-1} + b^{-1}\,a\,g\,d\,g\,i^{-1}\,c^{-1} + b^{-1}\,a\,g\,e\,z\,i^{-1}\,c^{-1} - e^{-1}\,d\,g\,d\,g\,i^{-1}\,c^{-1} - e^{-1}\,d\,g\,e\,z\,i^{-1}\,c^{-1} + e^{-1}\,i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1} + e^{-1}\,i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1} + b^{-1}\,a\,g\,i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1} + b^{-1}\,a\,g\,i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1} - e^{-1}\,d\,g\,i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1}$

$i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1}\,k^{-1}\,z\,d \to d + i^{-1}\,h\,a - d\,g\,i^{-1}\,c^{-1}\,k^{-1}\,j\,a - d\,g\,i^{-1}\,c^{-1}\,k^{-1}\,z\,d - e\,z\,i^{-1}\,c^{-1}\,k^{-1}\,j\,a - e\,z\,i^{-1}\,c^{-1}\,k^{-1}\,z\,d - i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1}\,k^{-1}\,j\,a - i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1}\,k^{-1}\,z\,d - i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1}\,k^{-1}\,j\,a$

$i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1}\,k^{-1}\,z\,e \;\to\; e + i^{-1}\,h\,b + d\,g\,i^{-1}\,c^{-1}\,k^{-1} + e\,z\,i^{-1}\,c^{-1}\,k^{-1} - d\,g\,i^{-1}\,c^{-1}\,k^{-1}\,j\,b - d\,g\,i^{-1}\,c^{-1}\,k^{-1}\,z\,e - e\,z\,i^{-1}\,c^{-1}\,k^{-1}\,j\,b - e\,z\,i^{-1}\,c^{-1}\,k^{-1}\,z\,e + i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1}\,k^{-1} + i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1}\,k^{-1} - i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1}\,k^{-1}\,j\,b - i^{-1}\,h\,a\,g\,i^{-1}\,c^{-1}\,k^{-1}\,z\,e - i^{-1}\,h\,b\,z\,i^{-1}\,c^{-1}\,k^{-1}\,j\,b$

$k^{-1}\,z\,d\,f\,k^{-1}\,j\,a\,d^{-1}\,e \;\to\; -1\,c\,h\,b - k^{-1}\,j\,b - k^{-1}\,z\,e + c\,h\,a\,d^{-1}\,e - k^{-1}\,j\,a\,f\,k^{-1} + k^{-1}\,j\,b\,j\,b - k^{-1}\,z\,d\,f\,k^{-1} + k^{-1}\,z\,e\,j\,b + k^{-1}\,j\,a\,f\,k^{-1}\,j\,b - k^{-1}\,j\,b\,j\,a\,d^{-1}\,e + k^{-1}\,z\,d\,f\,k^{-1}\,j\,b - k^{-1}\,z\,e\,j\,a\,d^{-1}\,e - k^{-1}\,j\,a\,f\,k^{-1}\,j\,a\,d^{-1}\,e$

$k^{-1}\,z\,e\,j\,a\,f\,k^{-1}\,z\,d \;\to\; -1\,c\,h\,a\,g\,d + c\,h\,b\,j\,a + k^{-1}\,j\,b\,j\,a + k^{-1}\,z\,d\,g\,d + c\,h\,a\,f\,k^{-1}\,j\,a + c\,h\,a\,f\,k^{-1}\,z\,d - k^{-1}\,j\,a\,f\,c\,h\,a + k^{-1}\,j\,a\,f\,k^{-1}\,j\,a + k^{-1}\,j\,a\,f\,k^{-1}\,z\,d - k^{-1}\,j\,a\,g\,d\,g\,d + k^{-1}\,j\,a\,g\,e\,j\,a + k^{-1}\,j\,b\,j\,a\,g\,d - k^{-1}\,j\,b\,j\,b\,j\,a - k^{-1}\,z\,d\,f\,c\,h\,a - k^{-1}\,z\,d\,g\,d\,g\,d + k^{-1}\,z\,d\,g\,e\,j\,a + k^{-1}\,z\,e\,j\,a\,g\,d - k^{-1}\,z\,e\,j\,b\,j\,a + k^{-1}\,j\,a\,g\,d\,f\,k^{-1}\,j\,a + k^{-1}\,j\,a\,g\,d\,f\,k^{-1}\,z\,d - k^{-1}\,j\,b\,j\,a\,f\,k^{-1}\,j\,a - k^{-1}\,j\,b\,j\,a\,f\,k^{-1}\,z\,d + k^{-1}\,z\,d\,g\,d\,f\,k^{-1}\,j\,a + k^{-1}\,z\,d\,g\,d\,f\,k^{-1}\,z\,d - k^{-1}\,z\,e\,j\,a\,f\,k^{-1}\,j\,a$

$k^{-1}\,z\,e\,j\,a\,f\,k^{-1}\,z\,e \;\to\; -1\,c\,h\,b - k^{-1}\,j\,b - c\,h\,a\,f\,k^{-1} - c\,h\,a\,g\,e + c\,h\,b\,j\,b - k^{-1}\,j\,a\,f\,k^{-1} - k^{-1}\,j\,a\,g\,e + 2\,k^{-1}\,j\,b\,j\,b + k^{-1}\,z\,e\,j\,b + c\,h\,a\,f\,k^{-1}\,j\,b + c\,h\,a\,f\,k^{-1}\,z\,e - k^{-1}\,j\,a\,f\,c\,h\,b + k^{-1}\,j\,a\,f\,k^{-1}\,j\,b + k^{-1}\,j\,a\,f\,k^{-1}\,z\,e - k^{-1}\,j\,a\,g\,d\,f\,k^{-1}$
$- k^{-1}\,j\,a\,g\,d\,g\,e + k^{-1}\,j\,a\,g\,e\,j\,b + k^{-1}\,j\,b\,j\,a\,f\,k^{-1} + k^{-1}\,j\,b\,j\,a\,g\,e - k^{-1}\,j\,b\,j\,b\,j\,b - k^{-1}\,z\,d\,f\,c\,h\,b - k^{-1}\,z\,d\,g\,d\,f\,k^{-1} - k^{-1}\,z\,d\,g\,d\,g\,e + k^{-1}\,z\,d\,g\,e\,j\,b + k^{-1}\,z\,e\,j\,a\,f\,k^{-1} + k^{-1}\,z\,e\,j\,a\,g\,e - k^{-1}\,z\,e\,j\,b\,j\,b + k^{-1}\,j\,a\,g\,d\,f\,k^{-1}\,j\,b + k^{-1}\,j\,a\,g\,d\,f\,k^{-1}\,z\,e - k^{-1}\,j\,b\,j\,a\,f\,k^{-1}\,j\,b - k^{-1}\,j\,b\,j\,a\,f\,k^{-1}\,z\,e + k^{-1}\,z\,d\,g\,d\,f\,k^{-1}\,j\,b + k^{-1}\,z\,d\,g\,d\,f\,k^{-1}\,z\,e - k^{-1}\,z\,e\,j\,a\,f\,k^{-1}\,j\,b$

$k^{-1}\,z\,e\,j\,b\,j\,b\,z\,i^{-1} \to c\,h\,a\,f\,c + c\,h\,a\,g\,i^{-1} - k^{-1}\,z\,d\,f\,c - k^{-1}\,z\,d\,g\,i^{-1} - c\,h\,a\,g\,d\,g\,i^{-1} - c\,h\,a\,g\,e\,z\,i^{-1} + c\,h\,b\,j\,a\,g\,i^{-1} + c\,h\,b\,j\,b\,z\,i^{-1} + k^{-1}\,j\,a\,g\,d\,f\,c + k^{-1}\,j\,a\,g\,d\,g\,i^{-1} - k^{-1}\,j\,b\,j\,a\,f\,c + k^{-1}\,j\,b\,j\,b\,z\,i^{-1} + k^{-1}\,z\,d\,g\,d\,f\,c + 2\,k^{-1}\,z\,d\,g\,d\,g\,i^{-1} + k^{-1}\,z\,d\,g\,e\,z\,i^{-1} - k^{-1}\,z\,e\,j\,a\,f\,c - k^{-1}\,z\,e\,j\,a\,g\,i^{-1} - k^{-1}\,j\,a\,f\,c\,h\,a\,g\,i^{-1} - k^{-1}\,j\,a\,f\,c\,h\,b\,z\,i^{-1} - k^{-1}\,j\,a\,g\,d\,g\,d\,g\,i^{-1} - k^{-1}\,j\,a\,g\,d\,g\,e\,z\,i^{-1} + k^{-1}\,j\,a\,g\,e\,j\,a\,g\,i^{-1} + k^{-1}\,j\,a\,g\,e\,j\,b\,z\,i^{-1} + k^{-1}\,j\,b\,j\,a\,g\,d\,g\,i^{-1} + k^{-1}\,j\,b\,j\,a\,g\,e\,z\,i^{-1} - k^{-1}\,j\,b\,j\,b\,j\,a\,g\,i^{-1} - k^{-1}\,j\,b\,j\,b\,j\,b\,z\,i^{-1} - k^{-1}\,z\,d\,f\,c\,h\,a\,g\,i^{-1} - k^{-1}\,z\,d\,f\,c\,h\,b\,z\,i^{-1} - k^{-1}\,z\,d\,g\,d\,g\,d\,g\,i^{-1} - k^{-1}\,z\,d\,g\,d\,g\,e\,z\,i^{-1} + k^{-1}\,z\,d\,g\,e\,j\,a\,g\,i^{-1} + k^{-1}\,z\,d\,g\,e\,j\,b\,z\,i^{-1} + k^{-1}\,z\,e\,j\,a\,g\,d\,g\,i^{-1} + k^{-1}\,z\,e\,j\,a\,g\,e\,z\,i^{-1} - k^{-1}\,z\,e\,j\,b\,j\,a\,g\,i^{-1}$

The time for preliminaries was 0:00:01
The time for NCMakeGB 1 was 0:00:00
The time for Remove Redundant 1 was 0:00:00
The time for the main NCMakeGB was 0:00:05
The time for Remove Redundant 2 was 0:00:00
The time for reducing unknowns was 0:00:01
The time for clean up basis was 0:00:02
The time for SmallBasis was 0:00:01
The time for CreateCategories was 0:01:07
The time for NCCV was 0:00:00
The time for RegularOutput was 0:00:38
The time for everything so far was 0:01:58

3 Appendix 3 - Find A Smaller Basis

```
Input:

SetNonCommutative[a,b,c,d,e,f,g,h,i,w,x,y,z,j,u,t,k,v,
Inv[a],Inv[b],Inv[c],Inv[d],Inv[e],Inv[f],Inv[g],Inv[h],
Inv[i],Inv[j], Inv[k]  ];

(*Here are the relations which we like and would like to retain*)
hopepolys =   {-1 + a**Inv[a], -1 + b**Inv[b], -1 + c**Inv[c],
   -1 + d**Inv[d], -1 + e**Inv[e], -1 + f**Inv[f],
   -1 + g**Inv[g], -1 + h**Inv[h], -1 + i**Inv[i],
   -1 + j**Inv[j], 3-1 + k**Inv[k], -1 + Inv[a]**a,     -1 + Inv[b]**b, -1 +
Inv[c]**c, -1 + Inv[d]**d, -1 + Inv[e]**e,     -1 + Inv[f]**f, -1 + Inv[g]**g, -1
+ Inv[h]**h, -1 + Inv[i]**i,
   -1 + Inv[j]**j, -1 + Inv[k]**k, t + a**g**Inv[i] + b**z**Inv[i],
   w - Inv[i] + d**g**Inv[i] + e**z**Inv[i],
   -z + j**a**g + j**b**z - k**c**i + z**d**g + z**e**z,
   u + Inv[k]**j**a + Inv[k]**z**d, v - Inv[k] + Inv[k]**j**b + Inv[k]**z**e,
   x - Inv[a] - f**Inv[k]**j + g**d**Inv[a] - f**Inv[k]**z**d**Inv[a],
   y - Inv[c]**Inv[k]**j**a**f - Inv[c]**Inv[k]**j**b**k -
    Inv[c]**Inv[k]**z**d**f - Inv[c]**Inv[k]**z**e**k
}

(* Here are the relations resulting from the original "discovery" run *)
  allpolys = {-1 + a**Inv[a], -1 + b**Inv[b], -1 + c**Inv[c], -1 + d**Inv[d],
   -1 + e**Inv[e], -1 + f**Inv[f], -1 + g**Inv[g], -1 + h**Inv[h],
   -1 + i**Inv[i], -1 + j**Inv[j], -1 + k**Inv[k], -1 + Inv[a]**a,
   -1 + Inv[b]**b, -1 + Inv[c]**c, -1 + Inv[d]**d, -1 + Inv[e]**e,
   -1 + Inv[f]**f, -1 + Inv[g]**g, -1 + Inv[h]**h, -1 + Inv[i]**i,
   -1 + Inv[j]**j, -1 + Inv[k]**k, t + a**g**Inv[i] + b**z**Inv[i],
   w - Inv[i] + d**g**Inv[i] + e**z**Inv[i],
   -z + j**a**g + j**b**z - k**c**i + z**d**g + z**e**z,
   u + Inv[k]**j**a + Inv[k]**z**d, v - Inv[k] + Inv[k]**j**b + Inv[k]**z**e,
   -Inv[a]**Inv[h] - Inv[d]**Inv[i] + Inv[a]**b**j**Inv[h] -
   Inv[a]**b**z**Inv[i] - Inv[d]**e**j**Inv[h] + Inv[d]**e**z**Inv[i],
   -Inv[f]**Inv[a] - Inv[k]**Inv[b] + Inv[f]**g**d**Inv[a] -
   Inv[f]**g**e**Inv[b] - Inv[k]**z**d**Inv[a] + Inv[k]**z**e**Inv[b],
   x - Inv[a] - f**Inv[k]**j + g**d**Inv[a] - f**Inv[k]**z**d**Inv[a],
   y - Inv[c]**Inv[k]**j**a**f - Inv[c]**Inv[k]**j**b**k -
   Inv[c]**Inv[k]**z**d**f - Inv[c]**Inv[k]**z**e**k,
   Inv[k] + Inv[f]**g**e + Inv[f]**Inv[a]**b - Inv[k]**z**e -
   Inv[f]**g**d**Inv[a]**b + Inv[k]**z**d**Inv[a]**b,
   h**a**g**Inv[i]**Inv[c] + h**b**z**Inv[i]**Inv[c] +
   i**d**g**Inv[i]**Inv[c] + i**e**z**Inv[i]**Inv[c] -
   Inv[c]**Inv[k]**j**a**f - Inv[c]**Inv[k]**j**b**k -
   Inv[c]**Inv[k]**z**d**f - Inv[c]**Inv[k]**z**e**k,
```

```
-c**h**a**g**Inv[i] - c**h**b**z**Inv[i] - c**i**d**g**Inv[i] -
c**i**e**z**Inv[i] + Inv[k]**j**a**f**c + Inv[k]**j**b**k**c +
Inv[k]**z**d**f**c + Inv[k]**z**e**k**c,
b**j + a**f**Inv[k]**j - a**g**d**Inv[a] - a**g**Inv[i]**h -
b**z**Inv[i]**h + a**f**Inv[k]**z**d**Inv[a],
d**Inv[a] + e**j + Inv[i]**h + d**f**Inv[k]**j - d**g**d**Inv[a] -
d**g**Inv[i]**h - e**z**Inv[i]**h + d**f**Inv[k]**z**d**Inv[a],
-g**Inv[i] + f**Inv[k]**j**Inv[h] - g**d**Inv[a]**Inv[h] +
Inv[a]**b**j**Inv[h] - Inv[a]**b**z**Inv[i] +
f**Inv[k]**z**d**Inv[a]**Inv[h],
-h**a - i**d + h**a**g**d - h**b**j**a + i**d**g**d - i**e**j**a -
h**a**f**Inv[k]**j**a - h**a**f**Inv[k]**z**d + h**a**g**Inv[i]**h**a +
h**b**z**Inv[i]**h**a - i**d**f**Inv[k]**j**a - i**d**f**Inv[k]**z**d +
i**d**g**Inv[i]**h**a + i**e**z**Inv[i]**h**a,
h**a**f**Inv[k] + h**a**g**e - h**b**j**b + i**d**f**Inv[k] +
i**d**g**e - i**e**j**b - h**a**f**Inv[k]**j**b -
h**a**f**Inv[k]**z**e + h**a**g**Inv[i]**h**b + h**b**z**Inv[i]**h**b -
i**d**f**Inv[k]**j**b - i**d**f**Inv[k]**z**e + i**d**g**Inv[i]**h**b +
i**e**z**Inv[i]**h**b, -j**a**f**c - j**b**k**c - z**d**f**c -
z**e**k**c + k**c**h**a**g**Inv[i] + k**c**h**b**z**Inv[i] +
k**c**i**d**g**Inv[i] + k**c**i**e**z**Inv[i],
z**d - j**a**g**d + j**b**j**a - k**c**h**a - z**d**g**d + z**e**j**a +
j**a**f**Inv[k]**j**a + j**a**f**Inv[k]**z**d + z**d**f**Inv[k]**j**a +
z**d**f**Inv[k]**z**d, -j**b - j**a**f**Inv[k] - j**a**g**e +
j**b**j**b - k**c**h**b - z**d**f**Inv[k] - z**d**g**e + z**e**j**b +
j**a**f**Inv[k]**j**b + j**a**f**Inv[k]**z**e + z**d**f**Inv[k]**j**b +
z**d**f**Inv[k]**z**e, -a**f**c + b**z**Inv[i] + a**g**d**g**Inv[i] +
a**g**e**z**Inv[i] - b**j**a**g**Inv[i] - b**j**b**z**Inv[i] +
a**g**Inv[i]**h**a**g**Inv[i] + a**g**Inv[i]**h**b**z**Inv[i] +
b**z**Inv[i]**h**a**g**Inv[i] + b**z**Inv[i]**h**b**z**Inv[i],
-d**f**c - d**g**Inv[i] + d**g**d**g**Inv[i] + d**g**e**z**Inv[i] -
e**j**a**g**Inv[i] - e**j**b**z**Inv[i] - Inv[i]**h**a**g**Inv[i] -
Inv[i]**h**b**z**Inv[i] + d**g**Inv[i]**h**a**g**Inv[i] +
d**g**Inv[i]**h**b**z**Inv[i] + e**z**Inv[i]**h**a**g**Inv[i] +
e**z**Inv[i]**h**b**z**Inv[i],
Inv[k]**z - c**h**a**Inv[d] - Inv[k]**j**a**g - Inv[k]**z**d**g +
Inv[k]**j**a**f**Inv[k]**z + Inv[k]**j**b**j**a**Inv[d] +
Inv[k]**z**d**f**Inv[k]**z + Inv[k]**z**e**j**a**Inv[d] +
Inv[k]**j**a**f**Inv[k]**j**a**Inv[d] +
Inv[k]**z**d**f**Inv[k]**j**a**Inv[d],
-Inv[k]**z + c**h**a**Inv[d] - c**h**b**Inv[e] - Inv[k]**j**b**Inv[e] -
Inv[k]**j**a**f**Inv[k]**Inv[e] - Inv[k]**j**b**j**a**Inv[d] +
Inv[k]**j**b**j**b**Inv[e] - Inv[k]**z**d**f**Inv[k]**Inv[e] -
Inv[k]**z**e**j**a**Inv[d] + Inv[k]**z**e**j**b**Inv[e] -
Inv[k]**j**a**f**Inv[k]**j**a**Inv[d] +
Inv[k]**j**a**f**Inv[k]**j**b**Inv[e]
- Inv[k]**z**d**f**Inv[k]**j**a**Inv[d] +
Inv[k]**z**d**f**Inv[k]**j**b**Inv[e],
z**Inv[i]**Inv[c] - Inv[b]**a**f - j**a**g**Inv[i]**Inv[c] -
```

```
j**b**z**Inv[i]**Inv[c] + z**Inv[i]**h**a**g**Inv[i]**Inv[c] +
z**Inv[i]**h**b**z**Inv[i]**Inv[c] + Inv[b]**a**g**d**g**Inv[i]**Inv[c] +
Inv[b]**a**g**e**z**Inv[i]**Inv[c] +
Inv[b]**a**g**Inv[i]**h**a**g**Inv[i]**Inv[c] +
Inv[b]**a**g**Inv[i]**h**b**z**Inv[i]**Inv[c],
-z**Inv[i]**Inv[c] + Inv[b]**a**f - Inv[e]**d**f -
Inv[e]**d**g**Inv[i]**Inv[c] - Inv[b]**a**g**d**g**Inv[i]**Inv[c] -
Inv[b]**a**g**e**z**Inv[i]**Inv[c] + Inv[e]**d**g**d**g**Inv[i]**Inv[c] +
Inv[e]**d**g**e**z**Inv[i]**Inv[c] -
Inv[e]**Inv[i]**h**a**g**Inv[i]**Inv[c] -
Inv[e]**Inv[i]**h**b**z**Inv[i]**Inv[c] -
Inv[b]**a**g**Inv[i]**h**a**g**Inv[i]**Inv[c] -
Inv[b]**a**g**Inv[i]**h**b**z**Inv[i]**Inv[c] +
Inv[e]**d**g**Inv[i]**h**a**g**Inv[i]**Inv[c] +
Inv[e]**d**g**Inv[i]**h**b**z**Inv[i]**Inv[c],
-d - Inv[i]**h**a + d**g**Inv[i]**Inv[c]**Inv[k]**j**a +
d**g**Inv[i]**Inv[c]**Inv[k]**z**d + e**z**Inv[i]**Inv[c]**Inv[k]**j**a +
e**z**Inv[i]**Inv[c]**Inv[k]**z**d +
Inv[i]**h**a**g**Inv[i]**Inv[c]**Inv[k]**j**a +
Inv[i]**h**a**g**Inv[i]**Inv[c]**Inv[k]**z**d +
Inv[i]**h**b**z**Inv[i]**Inv[c]**Inv[k]**j**a +
Inv[i]**h**b**z**Inv[i]**Inv[c]**Inv[k]**z**d,
-e - Inv[i]**h**b - d**g**Inv[i]**Inv[c]**Inv[k] -
e**z**Inv[i]**Inv[c]**Inv[k] + d**g**Inv[i]**Inv[c]**Inv[k]**j**b +
d**g**Inv[i]**Inv[c]**Inv[k]**z**e + e**z**Inv[i]**Inv[c]**Inv[k]**j**b +
e**z**Inv[i]**Inv[c]**Inv[k]**z**e -
Inv[i]**h**a**g**Inv[i]**Inv[c]**Inv[k] -
Inv[i]**h**b**z**Inv[i]**Inv[c]**Inv[k] +
Inv[i]**h**a**g**Inv[i]**Inv[c]**Inv[k]**j**b +
Inv[i]**h**a**g**Inv[i]**Inv[c]**Inv[k]**z**e +
Inv[i]**h**b**z**Inv[i]**Inv[c]**Inv[k]**j**b +
Inv[i]**h**b**z**Inv[i]**Inv[c]**Inv[k]**z**e,
c**h**b + Inv[k]**j**b + Inv[k]**z**e - c**h**a**Inv[d]**e +
Inv[k]**j**a**f**Inv[k] - Inv[k]**j**b**j**b + Inv[k]**z**d**f**Inv[k] -
Inv[k]**z**e**j**b - Inv[k]**j**a**f**Inv[k]**j**b +
Inv[k]**j**b**j**a**Inv[d]**e - Inv[k]**z**d**f**Inv[k]**j**b +
Inv[k]**z**e**j**a**Inv[d]**e + Inv[k]**j**a**f**Inv[k]**j**a**Inv[d]**e +
Inv[k]**z**d**f**Inv[k]**j**a**Inv[d]**e,
c**h**a**g**d - c**h**b**j**a - Inv[k]**j**b**j**a - Inv[k]**z**d**g**d -
c**h**a**f**Inv[k]**j**a - c**h**a**f**Inv[k]**z**d +
Inv[k]**j**a**f**c**h**a - Inv[k]**j**a**f**Inv[k]**j**a -
Inv[k]**j**a**f**Inv[k]**z**d + Inv[k]**j**a**g**d**g**d -
Inv[k]**j**a**g**e**j**a - Inv[k]**j**b**j**a**g**d +
Inv[k]**j**b**j**b**j**a + Inv[k]**z**d**f**c**h**a +
Inv[k]**z**d**g**d**g**d - Inv[k]**z**d**g**e**j**a -
Inv[k]**z**e**j**a**g**d + Inv[k]**z**e**j**b**j**a -
Inv[k]**j**a**g**d**f**Inv[k]**j**a - Inv[k]**j**a**g**d**f**Inv[k]**z**d +
Inv[k]**j**b**j**a**f**Inv[k]**j**a + Inv[k]**j**b**j**a**f**Inv[k]**z**d -
Inv[k]**z**d**g**d**f**Inv[k]**j**a - Inv[k]**z**d**g**d**f**Inv[k]**z**d +
Inv[k]**z**e**j**a**f**Inv[k]**j**a + Inv[k]**z**e**j**a**f**Inv[k]**z**d,
```

```
   c**h**b + Inv[k]**j**b + c**h**a**f**Inv[k] + c**h**a**g**e -
   c**h**b**j**b + Inv[k]**j**a**f**Inv[k] + Inv[k]**j**a**g**e -
   2*Inv[k]**j**b**j**b - Inv[k]**z**e**j**b - c**h**a**f**Inv[k]**j**b -
   c**h**a**f**Inv[k]**z**e + Inv[k]**j**a**f**c**h**b -
   Inv[k]**j**a**f**Inv[k]**j**b - Inv[k]**j**a**f**Inv[k]**z**e +
   Inv[k]**j**a**g**d**f**Inv[k] + Inv[k]**j**a**g**d**g**e -
   Inv[k]**j**a**g**e**j**b - Inv[k]**j**b**j**a**f**Inv[k] -
   Inv[k]**j**b**j**a**g**e + Inv[k]**j**b**j**b**j**b +
   Inv[k]**z**d**f**c**h**b + Inv[k]**z**d**g**d**f**Inv[k] +
   Inv[k]**z**d**g**d**g**e - Inv[k]**z**d**g**e**j**b -
   Inv[k]**z**e**j**a**f**Inv[k] - Inv[k]**z**e**j**a**g**e +
   Inv[k]**z**e**j**b**j**b - Inv[k]**j**a**g**d**f**Inv[k]**j**b -
   Inv[k]**j**a**g**d**f**Inv[k]**z**e + Inv[k]**j**b**j**a**f**Inv[k]**j**b +
   Inv[k]**j**b**j**a**f**Inv[k]**z**e - Inv[k]**z**d**g**d**f**Inv[k]**j**b -
   Inv[k]**z**d**g**d**f**Inv[k]**z**e +
   Inv[k]**z**e**j**a**f**Inv[k]**j**b + Inv[k]**z**e**j**a**f**Inv[k]**z**e,
   -c**h**a**f**c - c**h**a**g**Inv[i] + Inv[k]**z**d**f**c +
   Inv[k]**z**d**g**Inv[i] + c**h**a**g**d**g**Inv[i] +
   c**h**a**g**e**z**Inv[i] - c**h**b**j**a**g**Inv[i] -
   c**h**b**j**b**z**Inv[i] - Inv[k]**j**a**g**d**f**c -
   Inv[k]**j**a**g**d**g**Inv[i] + Inv[k]**j**b**j**a**f**c -
   Inv[k]**j**b**j**b**z**Inv[i] - Inv[k]**z**d**g**d**f**c -
   2*Inv[k]**z**d**g**d**g**Inv[i] - Inv[k]**z**d**g**e**z**Inv[i] +
   Inv[k]**z**e**j**a**f**c + Inv[k]**z**e**j**a**g**Inv[i] +
   Inv[k]**j**a**f**c**h**a**g**Inv[i] + Inv[k]**j**a**f**c**h**b**z**Inv[i] +
   Inv[k]**j**a**g**d**g**d**g**Inv[i] + Inv[k]**j**a**g**d**g**e**z**Inv[i] -
   Inv[k]**j**a**g**e**j**a**g**Inv[i] - Inv[k]**j**a**g**e**j**b**z**Inv[i] -
   Inv[k]**j**b**j**a**g**d**g**Inv[i] - Inv[k]**j**b**j**a**g**e**z**Inv[i] +
   Inv[k]**j**b**j**b**j**a**g**Inv[i] + Inv[k]**j**b**j**b**j**b**z**Inv[i] +
   Inv[k]**z**d**f**c**h**a**g**Inv[i] + Inv[k]**z**d**f**c**h**b**z**Inv[i] +
   Inv[k]**z**d**g**d**g**d**g**Inv[i] + Inv[k]**z**d**g**d**g**e**z**Inv[i] -
   Inv[k]**z**d**g**e**j**a**g**Inv[i] - Inv[k]**z**d**g**e**j**b**z**Inv[i] -
   Inv[k]**z**e**j**a**g**d**g**Inv[i] - Inv[k]**z**e**j**a**g**e**z**Inv[i] +
   Inv[k]**z**e**j**b**j**a**g**Inv[i] + Inv[k]**z**e**j**b**j**b**z**Inv[i]}

(* Here we set the order to be strictly graded lex *)

SetKnowns[a,Inv[a],b,Inv[b],c,Inv[c],d,Inv[d],e, Inv[e], f, Inv[f], g, Inv[g], h,
Inv[h],i, Inv[i],j, Inv[j], k, Inv[k] , z, y,x,t,u,v,w];

SetUnknowns[{}];

(* Ask for a small basis retaining the relations we like *)
SmallBasis[ allpolys, hopepolys, 3 ]
```

Output:

$$
\begin{aligned}
\{&-\mathrm{Inv}[a]**\mathrm{Inv}[h]-\mathrm{Inv}[d]**\mathrm{Inv}[i]+\mathrm{Inv}[a]**b**j**\mathrm{Inv}[h]-\\
&\quad \mathrm{Inv}[a]**b**z**\mathrm{Inv}[i]-\mathrm{Inv}[d]**e**j**\mathrm{Inv}[h]+\\
&\quad \mathrm{Inv}[d]**e**z**\mathrm{Inv}[i],\\
&-\mathrm{Inv}[f]**\mathrm{Inv}[a]-\mathrm{Inv}[k]**\mathrm{Inv}[b]+\\
&\quad \mathrm{Inv}[f]**g**d**\mathrm{Inv}[a]-\mathrm{Inv}[f]**g**e**\mathrm{Inv}[b]-\\
&\quad \mathrm{Inv}[k]**z**d**\mathrm{Inv}[a]+\mathrm{Inv}[k]**z**e**\mathrm{Inv}[b]\}
\end{aligned}
$$

4 Appendix 4 - Confirm Our Relations Imply The Result

```
Input:

SetNonCommutative[a,b,c,d,e,f,g,h,i,w,x,y,z,j,u,t,k,v,
Inv[a],Inv[b],Inv[c],Inv[d],Inv[e],Inv[f],Inv[g],Inv[h],
Inv[i],Inv[j], Inv[k]  ];

(* Here we create the relations which we wish to imply *)
first = {{a,t,b},{u,c,v},{d,w, e}};

second = {{ x,f,g}, {h,y,i},{j,k,z}};

oneway =MatMult[first,second] - IdentityMatrix[3];

otherway =
     MatMult[second,first]-  IdentityMatrix[3];

start = Flatten[{ oneway, otherway  }];

SetKnowns[a,Inv[a],b,Inv[b],c,Inv[c],d,Inv[d],e, Inv[e], f, Inv[f], g, Inv[g], h,
Inv[h],i, Inv[i],j, Inv[j], k, Inv[k] , z, y,x,t,u,v,w];

SetUnknowns[{}];

(* hopepolys = *)
(* hopepolys are defined as in Appendix 3 *)

(* Here are the relations found through the Small Basis algorithm *)

newrels = {-Inv[a]**Inv[h] - Inv[d]**Inv[i] + Inv[a]**b**j**Inv[h] -
   Inv[a]**b**z**Inv[i] - Inv[d]**e**j**Inv[h] + Inv[d]**e**z**Inv[i],
  -Inv[f]**Inv[a] - Inv[k]**Inv[b] + Inv[f]**g**d**Inv[a] -
   Inv[f]**g**e**Inv[b] - Inv[k]**z**d**Inv[a] + Inv[k]**z**e**Inv[b]}

hopepolys = Join[ hopepolys, newrels ];

(*  We will have our result if the starting relations are elements of
    the ideal generated by the above relations i.e. the GB created with
    the above relations reduces the starting relations to 0 *)

(*  Note that a 5 iteration GB failed to reduce the starting relations *)
hopeGB = NCMakeGB[ hopepolys , 7 ];

hoperules =  PolyToRule[ hopeGB ];

(* Use the Groebner basis relations to reduce the original relations *)
Reduction[ start , hoperules ]
```

Output:

$$\{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$$