Methods and programs for the generation of contextual finite geometries

Jessy Colonval

Franche-Comte university
FEMTO-ST Institute, DISC department, VESONTIO team
ANR project I-QUINS

jessy.colonval@femto-st.fr

Joint work with Henri de Boutray, Alain Giorgetti, Frédéric Holweck and Pierre-Alain Masson

November 28, 2019

Introduction - Context

- Master's degree in research supervised by A. Giorgetti.
- Project ANR I-SITE UBFC I-QUINS (*Integrated QUantum Information at the NanoScale*).
- Context: study of finite geometries called *quantum geometries* [PGHS15].
- With Magma Computational Algebra System [BCP97].

[PGHS15] M. Planat, A. Giorgetti, F. Holweck, M. Saniga.
Quantum contextual finite geometries from dessins d'enfants.
International Journal of Geometric Methods in Modern Physics.
2015.

[BCP97] W. Bosma, J. Canon, C. Playoust.
The Magma Algebra System I: The User Language.
Journal of Symbolic Computation.
1997.

Introduction - Contributions

- Implementation of a method for building finite geometries from Pauli groups [PS07].
- Implementation of a Kochen-Specker proof detection method [HS17].
- Implementation of a method for extracting critical Kochen-Specker proofs present in quantum finite geometry.

[PS07] M. Planat, M. Saniga.
On the Pauli graphs of N-qudits.
Quantum Information and Computation.
2007.

[HS17] F. Holweck, M. Saniga.
Contextuality with a Small Number of Observables.
International Journal of Quantum Information.
2017.

Introduction - Contributions

- Quantum geometries not constructed by the method [PS07] but is obtained by another process [PGHS15].
- Implementation of a correspondence between child's drawings and finite geometries [PGHS15].

[PS07] M. Planat, M. Saniga.
On the Pauli graphs of N-qudits.
Quantum Information and Computation.
2007.

[PGHS15] M. Planat, A. Giorgetti, F. Holweck, M. Saniga.
Quantum contextual finite geometries from dessins d'enfants.
International Journal of Geometric Methods in Modern Physics.
2015.

Block design (= finite geometry) and incidence structure

Definition (Block design)

A block is a non-empty part of a set Ω. A $\mathcal{B}$ block design is a set of blocks.

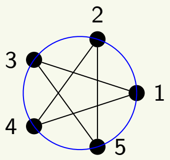
Definition (Incidence structure)

An incidence structure is a triplet $\mathcal{D} = (\Omega, \mathcal{B}, \mathcal{I})$ where $\Omega = \{1, \ldots, n\}$ is a set of finite elements, $\mathcal{B} = \{b_1, \ldots, b_p\}$ numbering a block design on Ω and $\mathcal{I} \subseteq \Omega \times \mathcal{B}$ is an incidence relationship, which defines membership of a element in a block.

Example of incidence structure

$\mathcal{I}$	1	2	3	4	5
b_1	1	1	1	1	1
b_2	1	0	1	0	0
b_3	1	0	0	1	0
b_4	0	1	0	1	0
b_5	0	1	0	0	1
b_6	0	0	1	0	1

$b_1 = \{1, 2, 3, 4, 5\}$
$b_2 = \{1, 3\}$
$b_3 = \{1, 4\}$
$b_4 = \{2, 4\}$
$b_5 = \{2, 5\}$
$b_6 = \{3, 5\}$



Vocabulary

MMP hypergraph	Block design	Finite geometry
k vertices	v elements	p points
m edges	b blocks	l lines
$\geq n$ vertices by edges	k elements by block	no constraint
edges intersect in at most $n-2$ vertices	each t-subset is in exactly λ blocks	no constraint

[PWMA19] M. Pavičić, M. Waegell, N. Megill, P.K. Aravind.
Automated generation of Kochen-Specker sets.
Scientific Reports.
2019.

[Col10] C. Colbourn.
CRC Handbook of Combinatorial Designs.
CRC Press.
2010.

Contents

The Pauli group

The matrix group composed of the four matrices

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ and } \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

is called the Pauli group of dimension 2, $\mathcal{P}_2$.

[PZ88] J. Patera, H. Zassenhaus.
The Pauli matrices in n dimensions and finest gradings of simple Lie algebras of type A_{n-1}.
Journal of Mathematical Physics.
1988.

Tensor products from the Pauli group

It is possible to generalize the Pauli group $\mathcal{P}_2$ to all dimensions $2^n \times 2^n$ from the tensor product of n Pauli's groups, $\mathcal{P}_2 \otimes \ldots \otimes \mathcal{P}_2$.

Definition (Tensor product)

Let A be a matrix of size $m \times n$ and B a matrix of size $p \times q$. Their tensor product is the matrix $A \otimes B$ of size mp by nq, defined by :

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \ldots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \ldots & a_{m,n}B \end{pmatrix}$$

Construction method

Proposition

Let $\mathcal{P}_n$ be a Pauli group of dimension n and a graph Γ where the vertices are matrices of $\mathcal{P}_n$ and the edges are present if two matrices are commutating ($A * B = B * A$). The finite geometry $G_{\mathcal{P}_n}$ is such that :

- *a vertex of $G_{\mathcal{P}_n}$ corresponds to a matrix of $\mathcal{P}_n$;*
- *the lines of $G_{\mathcal{P}_n}$ are the cliques of the graph Γ, i.e. the subsets of the vertices that form a complete graph.*

Implementation

```
/**
 * Computes incidence structures from groups of matrices.
 * For that, this fonction computes a graph where the vertices are matrices of the
 * group and the links are present if two matrices are commuting.
 * The geometry has for points the matrices of the group and for edges the cliques
 * of the graph [PS07].
 *
 * @param MatGrp::AlgMat A given group of matrices.
 * @return Inc The corresponding incidence structure.
 */
IncFromPauliGroup := function(MatGrp)
  nbGen := NumberOfGenerators(MatGrp);
  generators := {@ i : i in [1..nbGen] | not IsIdentity(MatGrp.i) @};
  edges := {};
  for i in generators do
    for j in generators do
      if i lt j and MatGrp.i * MatGrp.j eq MatGrp.j * MatGrp.i then
        Include(~edges, {i,j});
      end if;
    end for;
  end for;
  graph := Graph<generators | edges>;
  cliques := AllCliques(graph);
  idCliques := [{generators[Index(vertex)] : vertex in clique} : clique in cliques];
  return IncidenceStructure<generators | idCliques>;
end function;
```

Listing 1: Function of building a finite geometry from a Pauli group.

Example

Group $\mathcal{P}_{2^2}$

$$M_1=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}, M_2=\begin{pmatrix}0&-i&0&0\\i&0&0&0\\0&0&0&-i\\0&0&i&0\end{pmatrix}, M_3=\begin{pmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}, M_4=\begin{pmatrix}1&0&0&0\\0&-1&0&0\\0&0&1&0\\0&0&0&-1\end{pmatrix},$$

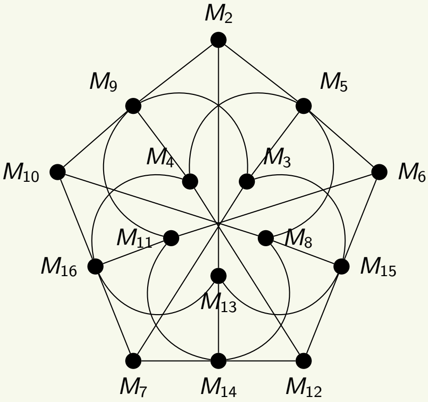
$$M_5=\begin{pmatrix}0&0&-i&0\\0&0&0&-i\\i&0&0&0\\0&i&0&0\end{pmatrix}, M_6=\begin{pmatrix}0&0&0&-1\\0&0&1&0\\0&1&0&0\\-1&0&0&0\end{pmatrix}, M_7=\begin{pmatrix}0&0&0&-i\\0&0&-i&0\\0&i&0&0\\i&0&0&0\end{pmatrix}, M_8=\begin{pmatrix}0&0&-i&0\\0&0&0&i\\i&0&0&0\\0&-i&0&0\end{pmatrix},$$

$$M_9=\begin{pmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{pmatrix}, M_{10}=\begin{pmatrix}0&0&0&-i\\0&0&i&0\\0&-i&0&0\\i&0&0&0\end{pmatrix}, M_{11}=\begin{pmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{pmatrix}, M_{12}=\begin{pmatrix}0&0&1&0\\0&0&0&-1\\1&0&0&0\\0&-1&0&0\end{pmatrix},$$

$$M_{13}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&-1\end{pmatrix}, M_{14}=\begin{pmatrix}0&-i&0&0\\i&0&0&0\\0&0&0&i\\0&0&-i&0\end{pmatrix}, M_{15}=\begin{pmatrix}0&1&0&0\\1&0&0&0\\0&0&0&-1\\0&0&-1&0\end{pmatrix}, M_{16}=\begin{pmatrix}1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&1\end{pmatrix}.$$

Example

The finite geometry $W(2)$ from $\mathcal{P}_{2^2}$



[PS07] M. Planat, M. Saniga.
On the Pauli graphs of N-qudits.
Quantum Information and Computation.
2007.

Contents

Kochen-Specker proofs

Proposition

A finite geometry of operators is a KS-proof if:

- *the lines of the configuration consist of mutually commuting operators, such a line is called a context;*
- *all operators square to identity;*
- *all operators belong to an even number of contexts;*
- *the product of the operators on the same context is* $\pm Id$*;*
- *there is an odd number of contexts giving* $-Id$*.*

[HS17] F. Holweck, M. Saniga.
Contextuality with a Small Number of Observables.
International Journal of Quantum Information.
2017.

Implementation

```
/**
 * Verifies that all squares of elements of a finite geometry are equal to the
 * identity matrix.
 *
 * @param MatGrp::AlgMat A given group of matrices.
 * @param I::Inc The corresponding incidence structure.
 * @return BoolElt corresponding to the satisfaction of property 2
 */
KSElementsCommuting := function(MatGrp, I)
  B := Blocks(I);
  for block in B do
    for idMat in Support(block) do
      if not IsIdentity(MatGrp.idMat^2) then
        return false;
      end if;
    end for;
  end for;
  return true;
end function;
```

Listing 2: Boolean function that verifies that all operator squares are equal to the identity matrix.

Implementation

```
/**
 * Checks if a finite geometry on matrices has each node in a even number on
 * lines
 *
 * @param MatGrp::AlgMat A given group of matrices.
 * @param I::Inc The corresponding incidence structure.
 * @return BoolElt corresponding to the satisfaction of property 3
 */
KSElementsContainmentParity := function(MatGrp, I)
  P := Points(I);
  for point in P do
    if (PointDegree(I, point) mod 2) eq 1 or PointDegree(I, point) eq 0 then
      return false;
    end if;
  end for;
  return true;
end function;
```

Listing 3: Boolean function that verifies that points of a finite geometry appear in an even number of lines.

Implementation

```
/**
 * Checks if a finite geometry on matrices has each product of elements on every
 * lines resulting to Id or -Id and if there is an odd number of lines resulting
 * to -Id
 *
 * @param MatGrp::AlgMat A given group of matrices.
 * @param I::Inc The corresponding incidence structure.
 * @return BoolElt corresponding to the satisfaction of properties 4 and 5
 */
KSLinesIdentity := function(MatGrp, I)
  B := Blocks(I);
  negCounter := 0;
  for block in B do
    res := 1;
    for idMat in Support(block) do
      res *:= MatGrp.idMat;
    end for;
    if not (IsIdentity(res) or IsIdentity(-res)) then
      return false;
    end if;
    if IsIdentity(-res) then
      negCounter +:= 1;
    end if;
  end for;
  if (negCounter mod 2) eq 0 then
    return false;
  end if;
  return true;
end function;
```

Listing 4: Boolean function that verifies that a finite geometry has an odd number of lines with an eigenvalue equal to -1.

Examples of not KS finite geometries

Eigenvalues of $W(2)$

	Lines	Eigenvalues (1 or -1)
b_1	$\{M_2, M_5, M_6\}$	1
b_2	$\{M_2, M_9, M_{10}\}$	1
b_3	$\{M_2, M_{13}, M_{14}\}$	1
b_4	$\{M_3, M_5, M_7\}$	1
b_5	$\{M_3, M_9, M_{11}\}$	1
b_6	$\{M_3, M_{13}, M_{15}\}$	1
b_7	$\{M_4, M_5, M_8\}$	1
b_8	$\{M_4, M_9, M_{12}\}$	1
b_9	$\{M_4, M_{13}, M_{16}\}$	1
b_{10}	$\{M_7, M_{10}, M_{16}\}$	1
b_{11}	$\{M_7, M_{12}, M_{14}\}$	-1
b_{12}	$\{M_8, M_{10}, M_{15}\}$	-1
b_{13}	$\{M_8, M_{11}, M_{14}\}$	1
b_{14}	$\{M_6, M_{11}, M_{16}\}$	-1
b_{15}	$\{M_6, M_{12}, M_{15}\}$	1

Examples of not KS finite geometries

Incidence structure of $W(2)$

Lines	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
b_1	1			1	1										
b_2	1							1	1						
b_3	1											1	1		
b_4		1		1		1									
b_5		1						1		1					
b_6		1										1		1	
b_7			1	1			1								
b_8			1					1			1				
b_9			1									1			1
b_{10}						1			1						1
b_{11}						1					1		1		
b_{12}							1		1					1	
b_{13}							1			1			1		
b_{14}					1					1					1
b_{15}					1						1			1	

Critical Kochen-Specker proof

Proposition

Let G be a finite geometry being a Kochen-Specker proof of and B the lines of G. Then critical Kochen-Specker's proofs are all the finite geometries G' respecting the properties of a Kochen-Specker proof and having a set of lines $B' \subset B$ such that $\nexists B'', B'' \subset B'$ who respecting the properties of a Kochen-Specker proof.

Implementation

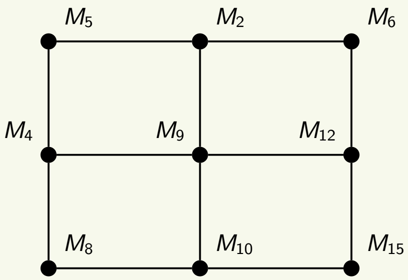
```
/**
 * Calculates the finite geometry from a group of Pauli matrices.
 * Remove the lines one by one and return all geometries that are
 * Kotchen-Specker proofs.
 *
 * @param MatGrp::AlgMat A given group of matrices.
 * @return SetEnum::allCriticalKS The list of the all incidence structures being
 *    Kochen-Specker's proofs.
 */
function KSOfInc(matGrp, I)
  B := { Support(block) : block in Blocks(I) };
  allKS := {};
  for i := 0 to #B do
    allInc:= SubInc(B, i);
    for inc in allInc do
      if KSProof(matGrp, inc) then
        Include(~allKS, inc);
      end if;
    end for;
  end for;
  return allKS;
end function;
```

Listing 5: Function that finds the critical Kochen-Specker proofs in a finite geometry.

Example

Critical KS of $W(2)$

For the finite geometry W(2) there are 10 critical Kochen-Specker proofs with 9 points and 6 lines, they are Mermin squares. One of 10 Mermin's squares is:



Contents

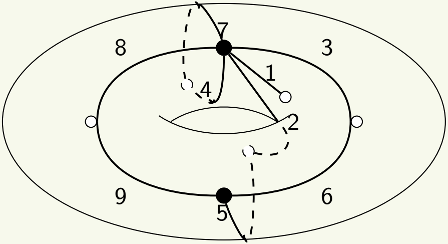
Child's drawing

Definition

A child's drawing is a bicolour map, drawn on an orientable surface, with all white vertices having a degree 1 or 2.

Example of child's drawing

$$\sigma = (1,2,4,8,7,3)(5,9,6)$$
$$\alpha = (2,5)(3,6)(4,7)(8,9)$$
$$\phi = (1,6,8,7,9,2)(3,4,5)$$



Construction method

Proposition

Let be $\mathcal{D}$ a child's drawing encoded by the permutation group P. Then the finished geometry $G_{\mathcal{D}}$ is such that [PGHS15]:

- *one point of $G_{\mathcal{D}}$ corresponds to a half edge of $\mathcal{D}$;*
- *all pairs of points in a line share the same stabilizer in P;*
- *the cardinality of the line stabilizer is the minimum value of all possible cardinalities.*

This construction allows to associate $G_{\mathcal{D}^i}$ finite geometries of n points to a child's drawing, where $i \in [2; n[$ represents the number of points per line.

[PGHS15] M. Planat, A. Giorgetti, F. Holweck, M. Saniga.
Quantum contextual finite geometries from dessins d'enfants.
International Journal of Geometric Methods in Modern Physics.
2015.

Implementation

```
/**
 * Verifies that all stabilizers of element pair are equal.
 *
 * @param grpPerm::GrpPerm A given permutations group.
 * @param line::SetEnum A given set of elements.
 * @return BoolElt Returns true if all stabilizers are equal else false.
 */
StabPairsAreEqual := function(grpPerm, line)
  allPairs := { Setseq(pair) : pair in Subsets(line, 2) };
  //Caution: different results between a set and a sequence of elements
  return forall(t){ <first, second> : first in allPairs, second in allPairs
    | Stabilizer(grpPerm, first) eq Stabilizer(grpPerm, second) };
end function;
```

Listing 6: Boolean function that indicates if the stabilizers of all point pairs are equal.

```
/**
 * Computes all stabilizer cardinalities.
 *
 * @param grpPerm::GrpPerm A given permutations group.
 * @return SeqEnum Sequence of all stabilizer cardinalities.
 */
ListCardStabilizers := function(grpPerm)
  nbElements := Degree(grpPerm);
  return [#Stabilizer(grpPerm, [1,e]) : e in [1..nbElements]];
end function;
```

Listing 7: Compute all possible stabilizer cardinalities.

Implementation

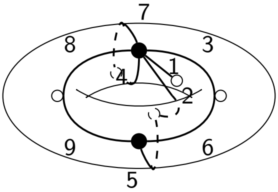
```
/**
 * Gives a finite geometry, if any, with a given cardinality of blocks from a
 * permutation group corresponding to a child drawing.
 *
 * @param PermGrp::GrpPerm A given permutation group.
 * @param card::RngIntElt The cardinality of blocks.
 * @return BlockDesign::SetEnum The corresponding block design.
 */
ChildDrawToFiniteGeoByStabMin := function(grpPerm, card)
  elements := Degree(grpPerm);
  SubSets := Subsets({1..elements}, card);
  blocks := { block : block in SubSets |
    CardStabBlockIsMin(grpPerm, block) and StabPairsAreEqual(grpPerm, block) };
  return IncidenceStructure<elements | blocks>;
end function;
```

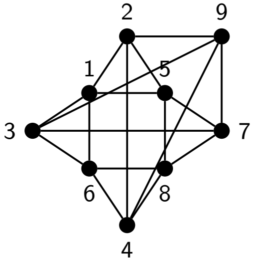
Listing 8: Method of constructing a finite geometry from a child's drawing.

Finite geometries found

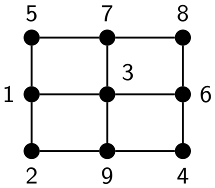
index	name	vertices	edges	number of points per line	occurrence
3	2-simplex (triangle)	3	3	2	3
4	3-simplex (tethahedron)	4	6	4	6
	square/quadrangle	4	4	2	4
5	4-simplex (5-cell)	5	10	2	15
6	5-simplex	6	15	2	31
	3-orthoplex (octahedron)	6	12	2	16
	bipartite graph $K(3,3)$	6	9	2	9
7	6-simplex	7	21	2	131
	Fano plane (7_3)	7	21	3	10
8	7-simplex	8	28	2	377
	4-orthoplex (16-cell)	8	24	2	51
	completed cube $K(4,4)$	8	16	2	54
9	8-simplex	9	36	2	1490
	Hesse $(9_4 1\underline{2}_3)$	9	36	3	14
	$K(3)^3$	9	27	2	60
	Pappus (9_3)	9	27	3	4
	(3×3)-grid	9	18	3	1
10	9-simplex	10	45	2	5277
	5-orthoplex	10	40	2	284
	bipartite graph $K(5,5)$	10	25	2	345
	Mermin's pentagram	10	30	4	3
	Desargues (10_3)	10	30	2	7

Example of finite geometry found





New finite geometry
(2 points per line)



Mermin's square
(3 points per line)

Contents

Conclusion - Not shown today

- Implementation of two Pauli group generalization methods for all dimensions [Kib09, PZ88].
- Implementation of a third method for constructing finite geometries from primitive permutation groups [Cd19].

[Kib09] M. Kibler.
An angular momentum approach to quadratic Fourier transform, Hadamard matrices, Gauss sums, mutually unbiased bases, the unitary group and the Pauli group.
Journal of Physics A: Mathematical and Theoretical.
2009.

[PZ88] J. Patera, H Zassenhaus.
The Pauli matrices in n dimensions and finest gradings of simple Lie algebras of type A_{n-1}.
Journal of Mathematical Physics.
1988.

[Cd19] J. Colonval, H. de Boutray.
Formalisation et validation d'une méthode de construction de systèmes de blocs.
18e journées Approches Formelles dans l'Assistance au Développement de Logiciels.
2019.

Conclusion - Future Work

- Use generation from child's drawings to identify geometries built from Pauli groups.
- Link geometry constructed from child's drawings to contextuality and determine if all generated geometries are Kochen-Specker proofs.

Thank you for your attention

Contents

Construction method

Proposition

Let G be a finite primitive permutation group acting on the set Ω of size n. Let $\alpha \in \Omega$, and let $\Delta \neq \{\alpha\}$ be an orbit of the stabilizer G_α of α. If

$$\mathcal{B} = \{\Delta^g : g \in G\}$$

and, given $\delta \in \Delta$,

$$\varepsilon = \{\{\alpha, \delta\}^g : g \in G\},$$

then $\mathcal{D} = (\Omega, \mathcal{B})$ forms a symmetric 1-$(n, |\Delta|, |\Delta|)$ design. Further, if Δ is a self-paired orbit of G_α then $\Gamma = (\Omega, \varepsilon)$ is a regular connected graph of valency $|\Delta|$, $\mathcal{D}$ is self-dual [...]

[KM08] J.D. Key, J. Moori.
Correction to: Codes, Designs and Graphs from the Janko Groups J_1 and J_2.
2008.

Orbit Δ calculation function

Proposition (Excerpt)

Let G be a finite primitive permutation group acting on the set Ω of size n. Let $\alpha \in \Omega$, and let $\Delta \neq \{\alpha\}$ be an orbit of the stabilizer G_α of α. [...]

```
/**
 * Compute orbits of stabilizers of a primitive group [KM02, Proposition 1].
 *
 * @param G::GrpPerm A primitive group
 * @return Deltas::Assoc An associative array indexed by alpha
 *    and containing the corresponding delta set
 */
AllDelta := function(G)
  n := Degree(G);
  Omega := {1..n};
  Deltas := AssociativeArray();
  for alpha in Omega do
    Galpha := Stabilizer(G, alpha);
    orbits := Orbits(Galpha);
    Deltas[alpha] := { IndexedSetToSet(Delta) : Delta in orbits | Delta ne { alpha } };
  end for;
  return Deltas;
end function;
```

Function for building block systems

Proposition (Excerpt)

Let G be a finite primitive permutation group acting on the set Ω of size n. [...]

$$\mathcal{B} = \{\Delta^g : g \in G\}[\ldots]$$

```
/**
 * Builds all block designs from a primitive group [KM02, Proposition 1]
 *
 * @param G::GrpPerm A primitive group
 * @return blocks::Assoc An associative array indexed by orbits delta
 *     and containing corresponding block designs
 */
BlckDsgnsFromPrmtvGrp := function(G)
  Deltas := AllDelta(G);
  blocks := AssociativeArray();
  for alpha in Keys(Deltas) do
    for Delta in Deltas[alpha] do
      blocks[Delta] := { Delta^g : g in G };
    end for;
  end for;
  return blocks;
end function;
```