GAP paketi za kombinatorne objekte*

Vedran Krčadinac

PMF-MO

24.2.2021.

* This work has been fully supported by the Croatian Science Foundation under the project 9752.

GAP i paketi

GAP – Groups, Algorithms, Programming

GAP is a system for computational discrete algebra, with particular emphasis on Computational Group Theory. The current release is GAP 4.11.0 and it can be obtained from our downloads page.

https://www.gap-system.org

GAP i paketi

GAP – Groups, Algorithms, Programming

GAP is a system for computational discrete algebra, with particular emphasis on Computational Group Theory. The current release is GAP 4.11.0 and it can be obtained from our downloads page.

https://www.gap-system.org

The GAP Group

Many people have helped in different ways to develop the GAP system, to maintain it, and to provide advice and support for users. All of these are refered to as the *GAP Group*.

GAP i paketi

GAP – Groups, Algorithms, Programming

GAP is a system for computational discrete algebra, with particular emphasis on Computational Group Theory. The current release is GAP 4.11.0 and it can be obtained from our downloads page.

https://www.gap-system.org

The GAP Group

Many people have helped in different ways to develop the GAP system, to maintain it, and to provide advice and support for users. All of these are refered to as the *GAP Group*.

A number of senior mathematicians and computer scientists have agreed to form a **GAP Council** that in particular functions as an editorial board for external contributions to GAP and will advise on broad policy issues in the continued development of GAP.

GAP i paketi

The present members of the GAP Council are:

Laurent Bartholdi, **Anton Betten**, Gene Cooperman, Bettina Eick, Graham Ellis, Gerhard Hiss, Derek Holt, Max Horn, Alexander Hulpke, Chris Jefferson, David Joyner, Alexander Konovalov, Steve Linton, Frank Luebeck, Max Neunhöffer, Alice Niemeyer, Götz Pfeiffer, Edmund Robertson, and **Leonard Soicher (chair)**.

GAP i paketi

The present members of the GAP Council are:

Laurent Bartholdi, **Anton Betten**, Gene Cooperman, Bettina Eick, Graham Ellis, Gerhard Hiss, Derek Holt, Max Horn, Alexander Hulpke, Chris Jefferson, David Joyner, Alexander Konovalov, Steve Linton, Frank Luebeck, Max Neunhöffer, Alice Niemeyer, Götz Pfeiffer, Edmund Robertson, and **Leonard Soicher (chair)**.

GAP Packages

Sets of user contributed programs, called **packages**, are distributed with GAP. For convenience of the GAP users, the GAP Group redistributes packages, but the package authors remain responsible for their maintenance.

GAP i paketi

We call a package an **accepted package** when it was successfully refereed or already distributed with GAP before the refereeing process was started. All other packages distributed here and not in this category are called **deposited packages**. These may be submitted for refereeing or the authors may not want to submit them.

GAP packages hosted on GitHub: http://gap-packages.github.io

GAP i paketi

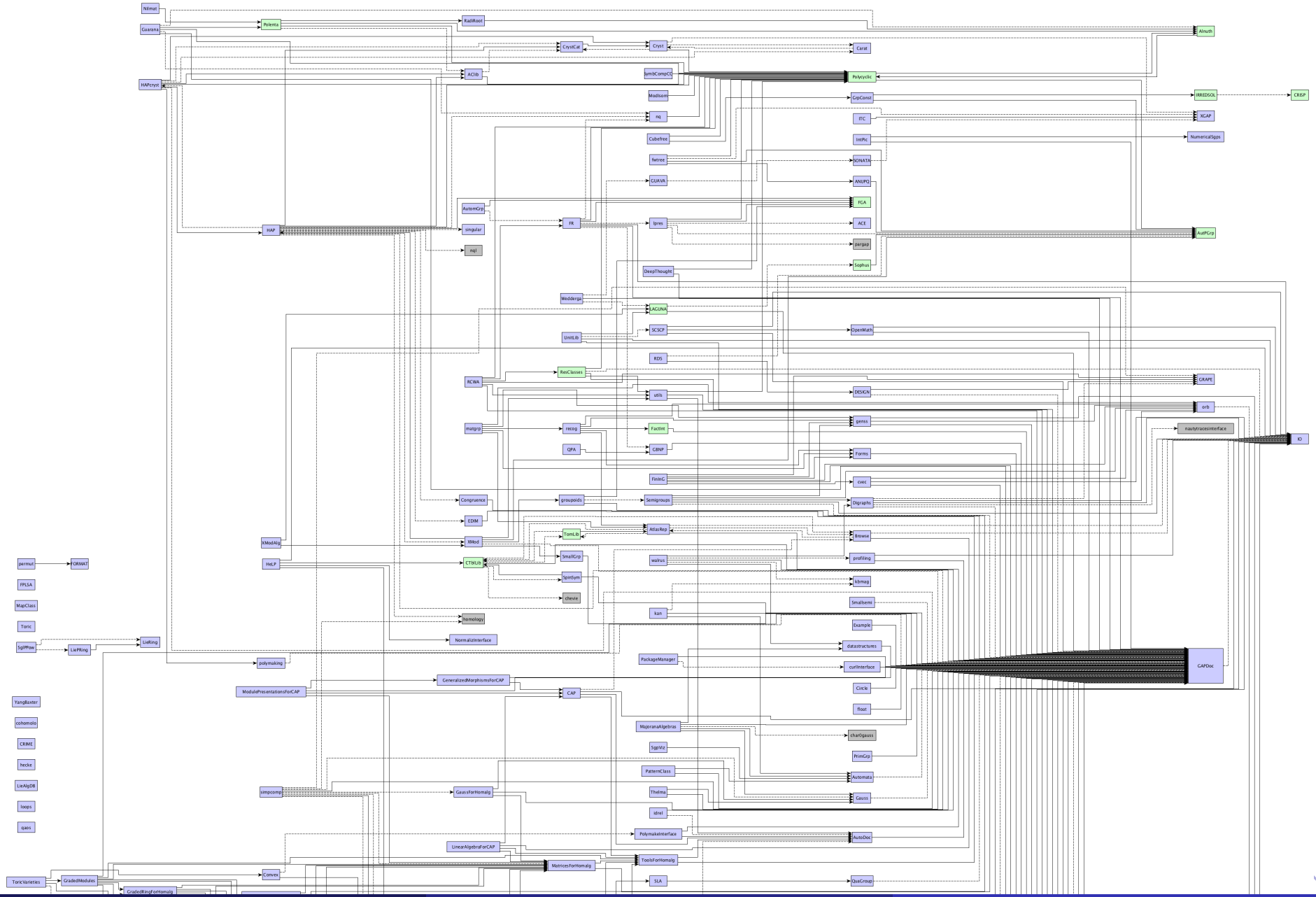
We call a package an **accepted package** when it was successfully refereed or already distributed with GAP before the refereeing process was started. All other packages distributed here and not in this category are called **deposited packages**. These may be submitted for refereeing or the authors may not want to submit them.

GAP packages hosted on GitHub: http://gap-packages.github.io

GAP paketi za kombinatorne objekte

- grafovi: **GRAPE**, **Digraphs**
- dizajni, konačne geometrije: **DESIGN**, **FinInG**, **UnitalSZ**
- diferencijski skupovi: **DifSets**, **RDS**

Međusobna ovisnost paketa



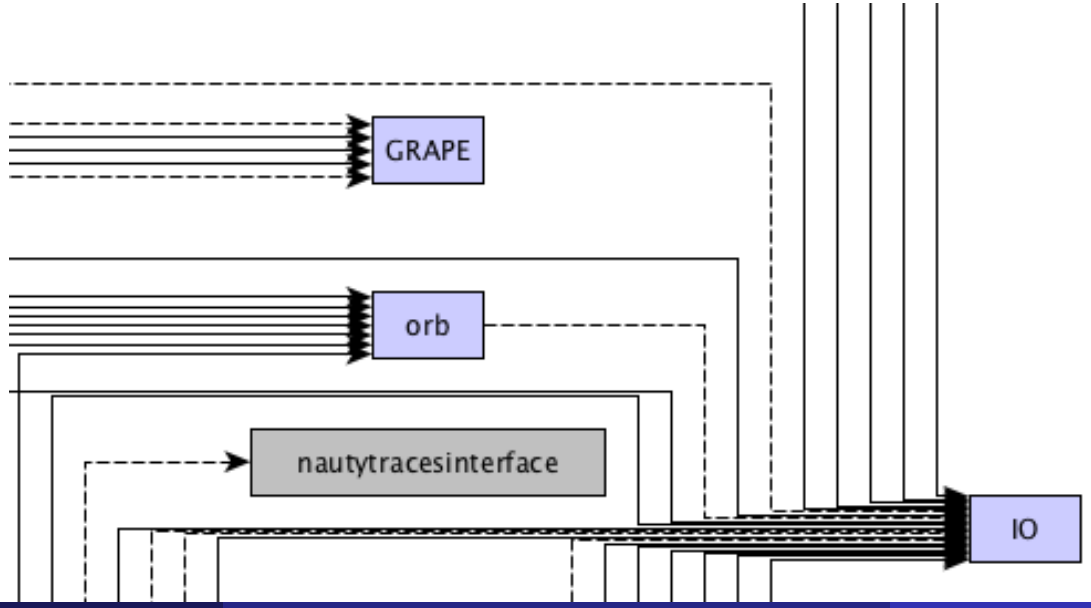
GRAPE – GRaph Algorithms using PErmutation groups

GRAPE is a package for computing with graphs and groups, and is primarily designed for constructing and analysing graphs related to groups, finite geometries, and designs.

Author & maintainer: **Leonard H. Soicher**

Status: accepted (01/07/1993)

Dependencies: GAP version $\geq$ 4.10



Digraphs

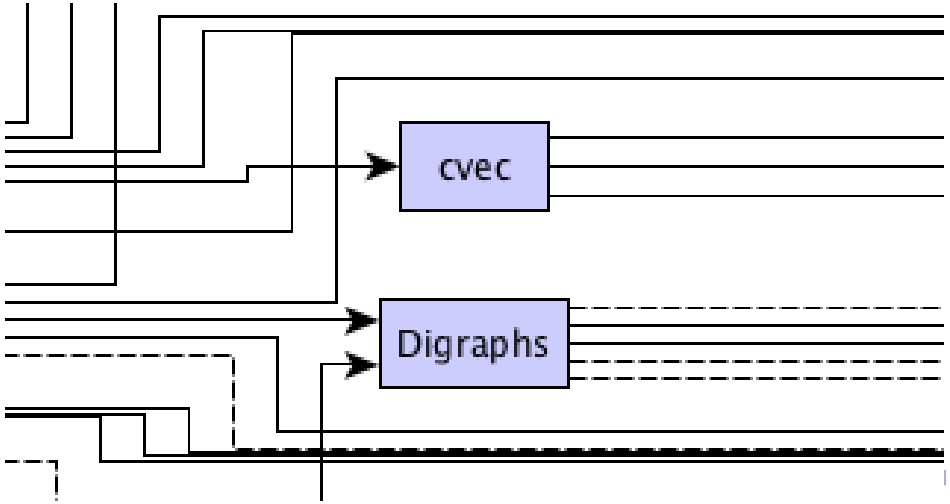
The Digraphs package is a GAP package for digraphs and multidigraphs.

Authors: **Jan De Beule**, Julius Jonusas, James Mitchell, Michael Torpey, Wilf Wilson

Maintainers: James Mitchell, Wilf Wilson

Status: deposited

Dependencies: needed - **IO**, **orb**, suggested - **GAPDoc**, **GRAPE**, **nautytracesinterface**



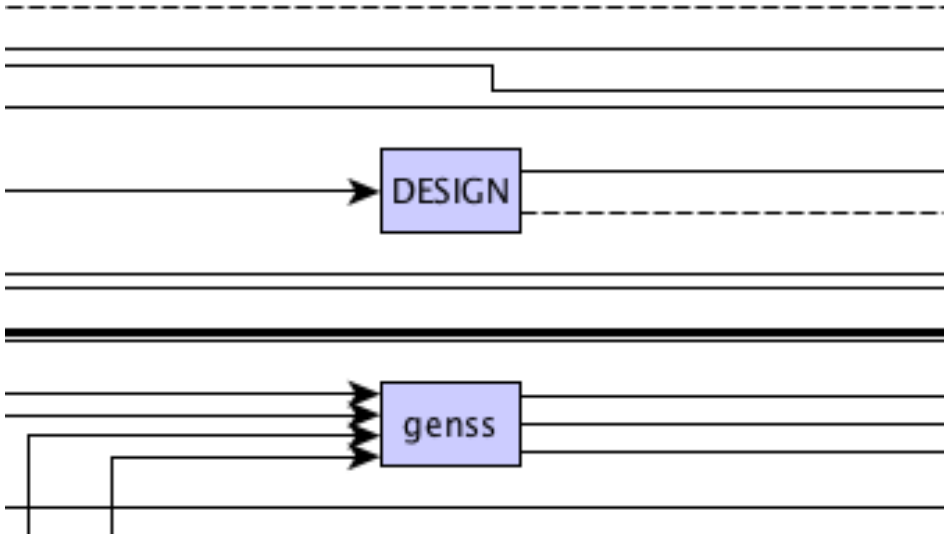
DESIGN

DESIGN is a package for constructing, classifying, partitioning, and studying block designs.

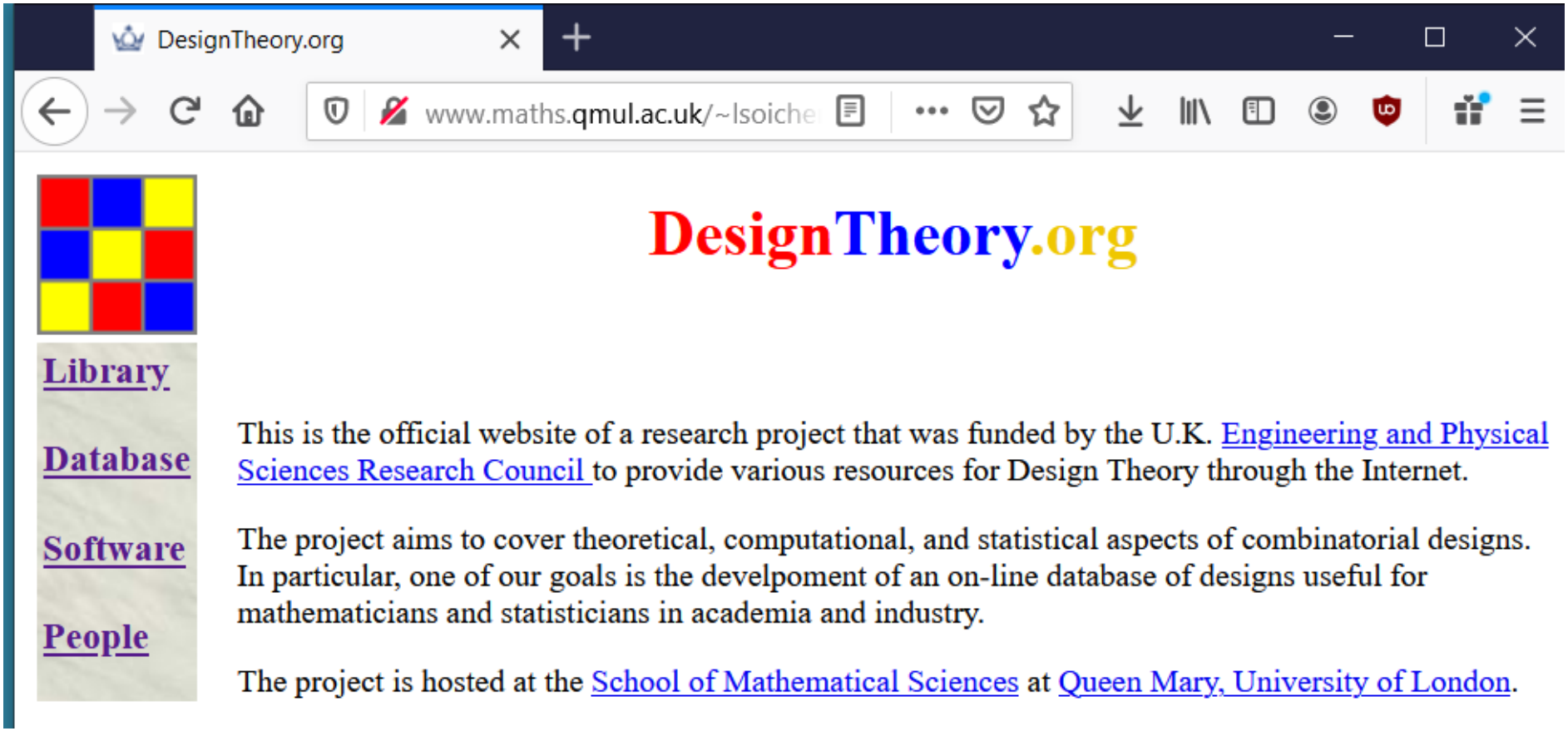
Author & maintainer: **Leonard H. Soicher**

Status: accepted (01/08/2006)

Dependencies: **GRAPE**, **GAPDoc**



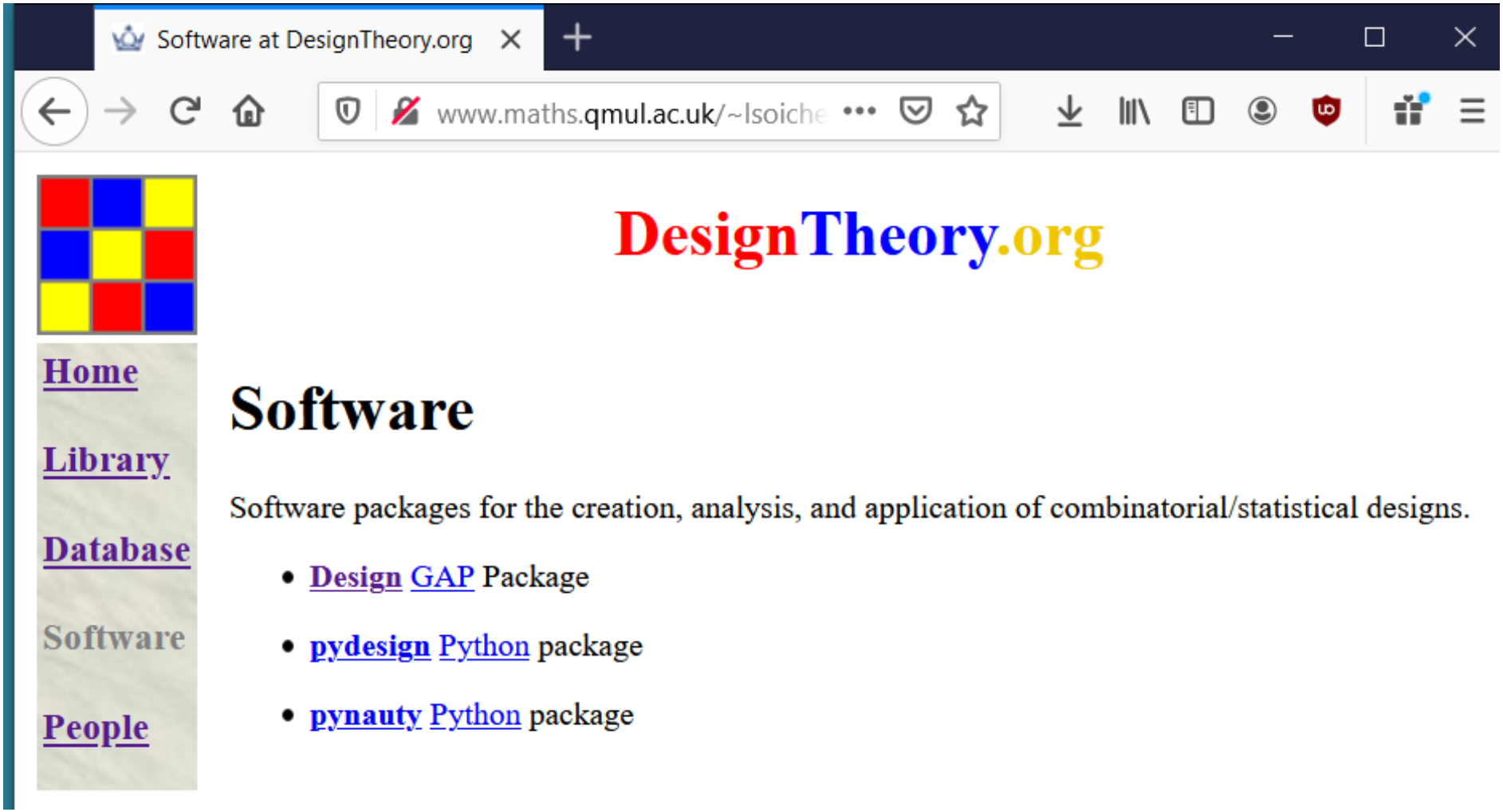
DesignTheory.org



R. A. Bailey, P. J. Cameron, P. Dobcsányi, J. P. Morgan, L. H. Soicher, *Designs on the Web*, Discrete Mathematics **306** (2006), 3014–3027.

https://doi.org/10.1016/j.disc.2004.10.027

DesignTheory.org



FinInG – Finite Incidence Geometry

FinInG is a package for computation in Finite Incidence Geometry. It provides users with the basic tools to work in various areas of finite geometry from the realms of projective spaces to the flat lands of generalised polygons. The algebraic power of GAP is employed, particularly in its facility with matrix and permutation groups.

Authors: John Bamberg, Anton Betten, Jan De Beule, Philippe Cara, Michel Lavrauw, Max Neunhöffer

Maintainers: John Bamberg, Jan De Beule, Philippe Cara, Michel Lavrauw

Status: accepted (01/11/2017)

Dependencies: **cvec**, **Forms**, **GAPDoc**, **genss**, **GRAPE**, **Orb**

UnitalSZ

This is a GAP package containing methods for abstract unitals as block designs. There are methods for automorphisms and isomorphisms and for the embeddings of unitals in the finite Desarguesian projective plane. There are functions for constructing unitals and some libraries of unitals of small order.

Authors: Gábor Péter Nagy, Dávid Mezőfi

Status: https://nagygp.github.io/UnitalSZ

Dependencies: **GAPDoc**, **Digraphs**, **io**

UnitalSZ

This is a GAP package containing methods for abstract unitals as block designs. There are methods for automorphisms and isomorphisms and for the embeddings of unitals in the finite Desarguesian projective plane. There are functions for constructing unitals and some libraries of unitals of small order.

Authors: Gábor Péter Nagy, Dávid Mezőfi

Status: https://nagygp.github.io/UnitalSZ

Dependencies: **GAPDoc**, **Digraphs**, **io**

https://nagygp.github.io
https://github.com/nagygp

Gábor P. Nagy

nagygp (Gábor P. Nagy) · GitHu | GAP package UnitalSZ | GitHub - nagygp/nagygp.githu | Loops

https://github.com/nagygp/

Why GitHub? Team Enterprise Explore Marketplace Pricing Search

Gábor P. Nagy

nagygp

Professor in Mathematics, research in Algebra and Geometry

GAP packages

Stable

- loops GAP package LOOPS
- UnitalSZ Abstract unitals and their designs
- BlissInterface Low level interface to the bliss graph automorphism tool,
- IncidenceStructures GAP implementation of abstract incidence structures

Experimental

- LRS4GAP LRS linear programming solver interface
- GZero Genus zero curves, divisors and codes
- OnAlgClosure Frobenius action on the algebraic closure
- Hermitian GAP package for Hermitian codes

DifSets

The DifSets package is a GAP package implementing an algorithm for enumerating all difference sets up to equivalence in a group.

Author & maintainer: Dylan Peifer

Status: accepted (01/07/2019)

Dependencies: needed - **GAPDoc**, **GRAPE**, suggested - **SmallGrp**

DifSets & RDS

DifSets

The DifSets package is a GAP package implementing an algorithm for enumerating all difference sets up to equivalence in a group.

Author & maintainer: Dylan Peifer

Status: accepted (01/07/2019)

Dependencies: needed - **GAPDoc**, **GRAPE**, suggested - **SmallGrp**

RDS

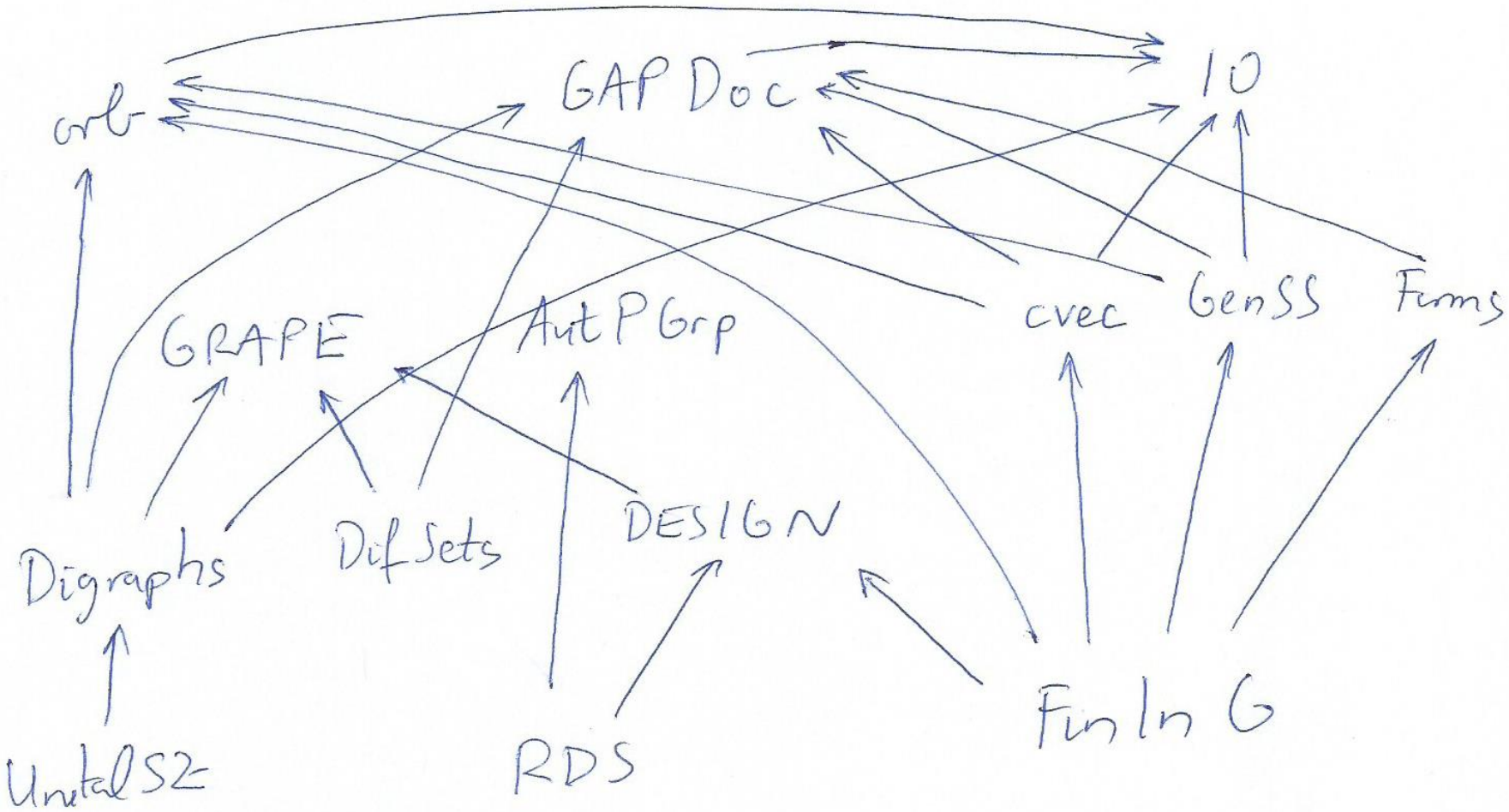
This package provides functions for the complete enumeration of relative difference sets.

Author & maintainer: Marc Roeder

Status: accepted (01/02/2008)

Dependencies: needed - **DESIGN**, suggested - **AutPGrp**

Međusobna ovisnost paketa



Još neki paketi

Matematički paketi

- AutPGrp
- genss
- Forms
- orb

Tehnički paketi

- cvec
- GAPDoc
- IO
- Example

AutPGrp – Computing the Automorphism Group of a p-Group

The AutPGrp package introduces a new function to compute the automorphism group of a finite p-group. The underlying algorithm is a refinement of the methods described in O'Brien (1995). In particular, this implementation is more efficient in both time and space requirements and hence has a wider range of applications than the ANUPQ method.

Authors: Bettina Eick, Eamonn O'Brien

Maintainers: Bettina Eick, Max Horn

Status: accepted (01/09/2000)

Dependencies: GAP version ≥ 4.7

genss – Generic Schreier-Sims

The genss package implements the randomised Schreier-Sims algorithm to compute a stabiliser chain and a base and strong generating set for arbitrary finite groups.

Authors: Max Neunhöffer, Felix Noeske

Maintainers: Felix Noeske, Max Horn

Status: deposited

Dependencies: **GAPDoc**, **IO**, **orb**

Forms & orb

Forms – Sesquilinear and Quadratic

This package can be used for work with sesquilinear and quadratic forms on finite vector spaces; objects that are used to describe polar spaces and classical groups.

Authors & Maintainers: John Bamberg, Jan De Beule

Status: accepted (01/03/2009)

Dependencies: **GAPDoc**

Forms & orb

Forms – Sesquilinear and Quadratic

This package can be used for work with sesquilinear and quadratic forms on finite vector spaces; objects that are used to describe polar spaces and classical groups.

Authors & Maintainers: John Bamberg, Jan De Beule

Status: accepted (01/03/2009)

Dependencies: **GAPDoc**

orb – Methods to enumerate orbits

The orb package is about enumerating orbits in various ways.

Authors: Juergen Mueller, Max Neunhöffer, Felix Noeske
Maintainers: Juergen Mueller, Felix Noeske, Max Horn

Status: deposited

Dependencies: **IO**

cvec – Compact vectors over finite fields

This package provides an implementation of compact vectors over finite fields. Contrary to earlier implementations no table lookups are used but only word-based processor arithmetic. This allows for bigger finite fields and higher speed.

Author: Max Neunhöffer

Maintainer: Max Horn

Status: deposited

Dependencies: **GAPDoc**, **IO**, **orb**

GAPDoc – A Meta Package for GAP Documentation

This package contains a definition of a structure for GAP (package) documentation, based on XML. It also contains conversion programs for producing text-, PDF- or HTML-versions of such documents, with hyperlinks if possible.

Authors: Frank Lübeck, Max Neunhöffer

Maintainer: Frank Lübeck

Status: accepted (01/10/2006)

Dependencies: **IO**.

External needs: (La)TeX installation for converting documents to PDF, BibTeX installation to produce unified labels for refs.

IO & Example

IO – Bindings for low level C library I/O routines

The IO package, as its name suggests, provides bindings for GAP to the lower levels of Input/Output functionality in the C library.

Author: Max Neunhöffer

Status: deposited

Maintainer: Max Horn

Dependencies: GAP version $\geq$ 4.10

IO & Example

IO – Bindings for low level C library I/O routines

The IO package, as its name suggests, provides bindings for GAP to the lower levels of Input/Output functionality in the C library.

Author: Max Neunhöffer

Status: deposited

Maintainer: Max Horn

Dependencies: GAP version ≥ 4.10

Example – Template of a GAP Package

The Example package, as its name suggests, is an example of how to create a GAP package. It has little functionality except for being a package.

Authors: Werner Nickel, Greg Gamble, Alexander Konovalov
Maintainer: Greg Gamble, Alexander Konovalov

Status: deposited

Dependencies: **GAPDoc**

The structure of a graph in GRAPE

In general GRAPE deals with finite directed graphs which may have loops but have no multiple edges. However, many GRAPE functions only work for simple graphs (i.e. no loops, and whenever $[x, y]$ is an edge then so is $[y, x]$), but these functions will check if an input graph is simple.

The structure of a graph in GRAPE

In general GRAPE deals with finite directed graphs which may have loops but have no multiple edges. However, many GRAPE functions only work for simple graphs (i.e. no loops, and whenever $[x, y]$ is an edge then so is $[y, x]$), but these functions will check if an input graph is simple.

In GRAPE, a graph *gamma* is stored as a **record**, with mandatory components `isGraph`, `order`, `group`, `schreierVector`, `representatives`, and `adjacencies`. Usually, the user need not be aware of this record structure, and is strongly advised only to use GRAPE functions to construct and modify graphs: *Graph*, *EdgeOrbitsGraph*, *NullGraph*, *CompleteGraph*, *JohnsonGraph*, *HammingGraph*, *CayleyGraph*, *GeneralizedOrbitalGraphs*, *AddEdgeOrbit*, *RemoveEdgeOrbit*, *AssignVertexNames*.

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

  GAP      GAP 4.11.0 of 29-Feb-2020
           https://www.gap-system.org
           Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> SymmetricGroup(5);
Sym( [ 1 .. 5 ] )
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> SymmetricGroup(5);
Sym( [ 1 .. 5 ] )
gap> Petersen2:=EdgeOrbitsGraph(a,[[1,3]],10);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  isSimple := true, order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> SymmetricGroup(5);
Sym( [ 1 .. 5 ] )
gap> Petersen2:=EdgeOrbitsGraph(a,[[1,3]],10);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  isSimple := true, order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> GraphIsomorphism(Petersen,Petersen2);
()
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> SymmetricGroup(5);
Sym( [ 1 .. 5 ] )
gap> Petersen2:=EdgeOrbitsGraph(a,[[1,3]],10);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  isSimple := true, order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> GraphIsomorphism(Petersen,Petersen2);
()
gap> Vertices(Petersen);
[ 1 .. 10 ]
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> SymmetricGroup(5);
Sym( [ 1 .. 5 ] )
gap> Petersen2:=EdgeOrbitsGraph(a,[[1,3]],10);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  isSimple := true, order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> GraphIsomorphism(Petersen,Petersen2);
()
gap> Vertices(Petersen);
[ 1 .. 10 ]
gap> Vertices(Petersen2);
[ 1 .. 10 ]
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> SymmetricGroup(5);
Sym( [ 1 .. 5 ] )
gap> Petersen2:=EdgeOrbitsGraph(a,[[1,3]],10);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  isSimple := true, order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> GraphIsomorphism(Petersen,Petersen2);
()
gap> Vertices(Petersen);
[ 1 .. 10 ]
gap> Vertices(Petersen2);
[ 1 .. 10 ]
gap> VertexNames(Petersen);
[ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ]
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
 GAP      https://www.gap-system.org
          Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> SymmetricGroup(5);
Sym( [ 1 .. 5 ] )
gap> Petersen2:=EdgeOrbitsGraph(a,[[1,3]],10);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  isSimple := true, order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> GraphIsomorphism(Petersen,Petersen2);
()
gap> Vertices(Petersen);
[ 1 .. 10 ]
gap> Vertices(Petersen2);
[ 1 .. 10 ]
gap> VertexNames(Petersen);
[ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ]
gap> VertexNames(Petersen2);
[ 1 .. 10 ]
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> Petersen2:=EdgeOrbitsGraph(a,[[1,3]],10);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  isSimple := true, order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> GraphIsomorphism(Petersen,Petersen2);
()
gap> Vertices(Petersen);
[ 1 .. 10 ]
gap> Vertices(Petersen2);
[ 1 .. 10 ]
gap> VertexNames(Petersen);
[ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ]
gap> VertexNames(Petersen2);
[ 1 .. 10 ]
gap> Petersen3:=ComplementGraph(JohnsonGraph(5,2));
rec( adjacencies := [ [ 8, 9, 10 ] ], group := Group([ (1,5,8,10,4)(2,6,9,3,7), (2,5)(3,6)(4,7) ]), isGraph := true,
  isSimple := true, names := [ [ 1, 2 ], [ 1, 3 ], [ 1, 4 ], [ 1, 5 ], [ 2, 3 ], [ 2, 4 ], [ 2, 5 ], [ 3, 4 ],
      [ 3, 5 ], [ 4, 5 ] ], order := 10, representatives := [ 1 ], schreierVector := [ -1, 2, 2, 1, 1, 1, 2, 1, 1, 1
    ] )
gap>
```

GRAPE – funkcije za kreiranje grafova

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
            CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
            Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
            Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
            Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> a:=Petersen.group;
Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ])
gap> Petersen2:=EdgeOrbitsGraph(a,[[1,3]],10);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  isSimple := true, order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> GraphIsomorphism(Petersen,Petersen2);
()
gap> Vertices(Petersen);
[ 1 .. 10 ]
gap> Vertices(Petersen2);
[ 1 .. 10 ]
gap> VertexNames(Petersen);
[ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ]
gap> VertexNames(Petersen2);
[ 1 .. 10 ]
gap> Petersen3:=ComplementGraph(JohnsonGraph(5,2));
rec( adjacencies := [ [ 8, 9, 10 ] ], group := Group([ (1,5,8,10,4)(2,6,9,3,7), (2,5)(3,6)(4,7) ]), isGraph := true,
  isSimple := true, names := [ [ 1, 2 ], [ 1, 3 ], [ 1, 4 ], [ 1, 5 ], [ 2, 3 ], [ 2, 4 ], [ 2, 5 ], [ 3, 4 ],
      [ 3, 5 ], [ 4, 5 ] ], order := 10, representatives := [ 1 ], schreierVector := [ -1, 2, 2, 1, 1, 1, 2, 1, 1, 1
     ] )
gap> GraphIsomorphism(Petersen,Petersen3);
(2,6,5,9)(3,8,10,7,4)
gap>
```

GAP struktura podataka **record** (zapis)

Records are next to lists the most important way to collect objects together. A record is a collection of components. Each component has a unique name, which is an identifier that distinguishes this component, and a value, which is an object of arbitrary type. We often abbreviate value of a component to element. We also say that a record contains its elements. You can access and change the elements of a record using its name.

GAP struktura podataka **record** (zapis)

Records are next to lists the most important way to collect objects together. A record is a collection of components. Each component has a unique name, which is an identifier that distinguishes this component, and a value, which is an object of arbitrary type. We often abbreviate value of a component to element. We also say that a record contains its elements. You can access and change the elements of a record using its name.

```
gap> r:=rec(a:=1,b:=2);
rec( a := 1, b := 2 )
gap> r.a;
1
gap> r.b;
2
gap> r.a:=7;
7
gap> r;
rec( a := 7, b := 2 )
```

GAP struktura podataka **record** (zapis)

We shall say that two records are identical if changing one of them by a record assignment also changes the other one. This is slightly incorrect, because if two records are identical, there are actually only two names for one record. However, the correct usage would be very awkward and would only add to the confusion. Note that two identical records must be equal, because there is only one records with two different names. Thus identity is an equivalence relation that is a refinement of equality.

GAP struktura podataka **record** (zapis)

We shall say that two records are identical if changing one of them by a record assignment also changes the other one. This is slightly incorrect, because if two records are identical, there are actually only two names for one record. However, the correct usage would be very awkward and would only add to the confusion. Note that two identical records must be equal, because there is only one records with two different names. Thus identity is an equivalence relation that is a refinement of equality.

```
gap> r1:=rec(a:=1,b:=2);
rec( a := 1, b := 2 )
gap> r2:=r1;
rec( a := 1, b := 2 )
gap> r3:=rec(a:=1,b:=2);
rec( a := 1, b := 2 )
```

```
gap> r2.a:=7;
7
gap> r1;
rec( a := 7, b := 2 )
gap> r3;
rec( a := 1, b := 2 )
```

GRAPE – struktura podataka za grafove

The `order` component contains the number of vertices of *gamma*. The vertices of *gamma* are always $1, 2, \ldots,$ *gamma*.`order`, but they may also be given names, either by a user (using *AssignVertexNames*) or by a function constructing a graph (e.g. *InducedSubgraph*, *BipartiteDouble*, *QuotientGraph*).

GRAPE – struktura podataka za grafove

The `order` component contains the number of vertices of *gamma*. The vertices of *gamma* are always $1, 2, \ldots,$ *gamma*.`order`, but they may also be given names, either by a user (using *AssignVertexNames*) or by a function constructing a graph (e.g. *InducedSubgraph*, *BipartiteDouble*, *QuotientGraph*).

The `names` component, if present, records these names, with *gamma*.`names`[i] the name of vertex i. If the `names` component is not present (the user may, for example, choose to unbind it), then the names are taken to be $1, 2, \ldots,$ *gamma*.`order`.

GRAPE – struktura podataka za grafove

The `order` component contains the number of vertices of *gamma*. The vertices of *gamma* are always $1, 2, \ldots,$ *gamma*.`order`, but they may also be given names, either by a user (using *AssignVertexNames*) or by a function constructing a graph (e.g. *InducedSubgraph*, *BipartiteDouble*, *QuotientGraph*).

The `names` component, if present, records these names, with *gamma*.`names`[i] the name of vertex i. If the `names` component is not present (the user may, for example, choose to unbind it), then the names are taken to be $1, 2, \ldots,$ *gamma*.`order`.

The `group` component records the GAP permutation group associated with *gamma* (this group must be a subgroup of the automorphism group of *gamma*).

GRAPE – struktura podataka za grafove

The `representatives` component records a set of orbit representatives for the action of *gamma*.`group` on the vertices of *gamma*, with *gamma*.`adjacencies`[*i*] being the set of vertices adjacent to *gamma*.`representatives`[*i*].

GRAPE – struktura podataka za grafove

The `representatives` component records a set of orbit representatives for the action of *gamma*.`group` on the vertices of *gamma*, with *gamma*.`adjacencies`[*i*] being the set of vertices adjacent to *gamma*.`representatives`[*i*].

The `group` and `schreierVector` components are used to compute the adjacency-set of an arbitrary vertex of *gamma* (this is done by the function *Adjacency*).

GRAPE – struktura podataka za grafove

The `representatives` component records a set of orbit representatives for the action of *gamma*.`group` on the vertices of *gamma*, with *gamma*.`adjacencies`[*i*] being the set of vertices adjacent to *gamma*.`representatives`[*i*].

The `group` and `schreierVector` components are used to compute the adjacency-set of an arbitrary vertex of *gamma* (this is done by the function *Adjacency*).

The only mandatory component which may change once a graph is initially constructed is `adjacencies` (when an edge-orbit of *gamma*.`group` is added to, or removed from, *gamma*). A graph record may also have some of the optional components `isSimple`, `autGroup`, `maximumClique`, `minimumVertexColouring`, and `canonicalLabelling`, which record information about that graph.

GRAPE – struktura podataka za grafove

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap>
```

GRAPE – struktura podataka za grafove

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap>
```

GRAPE – struktura podataka za grafove

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> AutomorphismGroup(Petersen);
Group([ (2,4)(6,9)(7,10), (2,6)(4,9)(5,8), (2,7)(3,5)(4,10)(6,9), (1,2,3,5,7)(4,6,8,9,10) ])
gap>
```

GRAPE – struktura podataka za grafove

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
            Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
            Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
            Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> AutomorphismGroup(Petersen);
Group([ (2,4)(6,9)(7,10), (2,6)(4,9)(5,8), (2,7)(3,5)(4,10)(6,9), (1,2,3,5,7)(4,6,8,9,10) ])
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  autGroup := Group( [ ( 2, 4)( 6, 9)( 7,10), (2,6)(4,9)(5,8), ( 2, 7)( 3, 5)( 4,10)( 6, 9),
      ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10) ] ),
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  isSimple := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap>
```

GRAPE – struktura podataka za grafove

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("GRAPE");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
true
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> AutomorphismGroup(Petersen);
Group([ (2,4)(6,9)(7,10), (2,6)(4,9)(5,8), (2,7)(3,5)(4,10)(6,9), (1,2,3,5,7)(4,6,8,9,10) ])
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  autGroup := Group( [ ( 2, 4)( 6, 9)( 7,10), (2,6)(4,9)(5,8), ( 2, 7)( 3, 5)( 4,10)( 6, 9),
      ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10) ] ),
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  isSimple := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> trokut:=UnderlyingGraph(EdgeOrbitsGraph(Group(()),[[1,2],[1,3],[2,3]],3));
rec( adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ], group := Group(()), isGraph := true, isSimple := true,
  order := 3, representatives := [ 1, 2, 3 ], schreierVector := [ -1, -2, -3 ] )
gap>
```

GRAPE – struktura podataka za grafove

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
gap> Petersen:=Graph(SymmetricGroup(5),[[1,2]],OnSets,function(x,y) return Intersection(x,y)=[]; end);
rec( adjacencies := [ [ 3, 5, 8 ] ], group := Group([ (1,2,3,5,7)(4,6,8,9,10), (2,4)(6,9)(7,10) ]), isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> AutomorphismGroup(Petersen);
Group([ (2,4)(6,9)(7,10), (2,6)(4,9)(5,8), (2,7)(3,5)(4,10)(6,9), (1,2,3,5,7)(4,6,8,9,10) ])
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  autGroup := Group( [ ( 2, 4)( 6, 9)( 7,10), (2,6)(4,9)(5,8), ( 2, 7)( 3, 5)( 4,10)( 6, 9),
      ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10) ] ),
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  isSimple := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> trokut:=UnderlyingGraph(EdgeOrbitsGraph(Group(()),[[1,2],[1,3],[2,3]],3));
rec( adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ], group := Group(()), isGraph := true, isSimple := true,
  order := 3, representatives := [ 1, 2, 3 ], schreierVector := [ -1, -2, -3 ] )
gap> Display(trokut);
rec(
  adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ],
  group := Group( [ () ] ),
  isGraph := true,
  isSimple := true,
  order := 3,
  representatives := [ 1, 2, 3 ],
  schreierVector := [ -1, -2, -3 ] )
gap>
```

GRAPE – struktura podataka za grafove

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10, representatives := [ 1 ], schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> AutomorphismGroup(Petersen);
Group([ (2,4)(6,9)(7,10), (2,6)(4,9)(5,8), (2,7)(3,5)(4,10)(6,9), (1,2,3,5,7)(4,6,8,9,10) ])
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  autGroup := Group( [ ( 2, 4)( 6, 9)( 7,10), (2,6)(4,9)(5,8), ( 2, 7)( 3, 5)( 4,10)( 6, 9),
      ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10) ] ),
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  isSimple := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> trokut:=UnderlyingGraph(EdgeOrbitsGraph(Group(()),[[1,2],[1,3],[2,3]],3));
rec( adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ], group := Group(()), isGraph := true, isSimple := true,
  order := 3, representatives := [ 1, 2, 3 ], schreierVector := [ -1, -2, -3 ] )
gap> Display(trokut);
rec(
  adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ],
  group := Group( [ () ] ),
  isGraph := true,
  isSimple := true,
  order := 3,
  representatives := [ 1, 2, 3 ],
  schreierVector := [ -1, -2, -3 ] )
gap> AutomorphismGroup(trokut);
Group([ (2,3), (1,2) ])
gap>
```

GRAPE – struktura podataka za grafove

/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

```
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> AutomorphismGroup(Petersen);
Group([ (2,4)(6,9)(7,10), (2,6)(4,9)(5,8), (2,7)(3,5)(4,10)(6,9), (1,2,3,5,7)(4,6,8,9,10) ])
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  autGroup := Group( [ ( 2, 4)( 6, 9)( 7,10), (2,6)(4,9)(5,8), ( 2, 7)( 3, 5)( 4,10)( 6, 9),
      ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10) ] ),
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  isSimple := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> trokut:=UnderlyingGraph(EdgeOrbitsGraph(Group(()),[[1,2],[1,3],[2,3]],3));
rec( adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ], group := Group(()), isGraph := true, isSimple := true,
  order := 3, representatives := [ 1, 2, 3 ], schreierVector := [ -1, -2, -3 ] )
gap> Display(trokut);
rec(
  adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ],
  group := Group( [ () ] ),
  isGraph := true,
  isSimple := true,
  order := 3,
  representatives := [ 1, 2, 3 ],
  schreierVector := [ -1, -2, -3 ] )
gap> AutomorphismGroup(trokut);
Group([ (2,3), (1,2) ])
gap> StructureDescription(last);
"S3"
gap>
```

GRAPE – struktura podataka za grafove

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
Group([ (2,4)(6,9)(7,10), (2,6)(4,9)(5,8), (2,7)(3,5)(4,10)(6,9), (1,2,3,5,7)(4,6,8,9,10) ])
gap> Display(Petersen);
rec(
  adjacencies := [ [ 3, 5, 8 ] ],
  autGroup := Group( [ ( 2, 4)( 6, 9)( 7,10), (2,6)(4,9)(5,8), ( 2, 7)( 3, 5)( 4,10)( 6, 9),
      ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10) ] ),
  group := Group( [ ( 1, 2, 3, 5, 7)( 4, 6, 8, 9,10), ( 2, 4)( 6, 9)( 7,10) ] ),
  isGraph := true,
  isSimple := true,
  names := [ [ 1, 2 ], [ 2, 3 ], [ 3, 4 ], [ 1, 3 ], [ 4, 5 ], [ 2, 4 ], [ 1, 5 ], [ 3, 5 ], [ 1, 4 ], [ 2, 5 ] ],
  order := 10,
  representatives := [ 1 ],
  schreierVector := [ -1, 1, 1, 2, 1, 1, 1, 1, 2, 2 ] )
gap> trokut:=UnderlyingGraph(EdgeOrbitsGraph(Group(()),[[1,2],[1,3],[2,3]],3));
rec( adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ], group := Group(()), isGraph := true, isSimple := true,
  order := 3, representatives := [ 1, 2, 3 ], schreierVector := [ -1, -2, -3 ] )
gap> Display(trokut);
rec(
  adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ],
  group := Group( [ () ] ),
  isGraph := true,
  isSimple := true,
  order := 3,
  representatives := [ 1, 2, 3 ],
  schreierVector := [ -1, -2, -3 ] )
gap> AutomorphismGroup(trokut);
Group([ (2,3), (1,2) ])
gap> StructureDescription(last);
"S3"
gap> Display(trokut);
rec(
  adjacencies := [ [ 2, 3 ], [ 1, 3 ], [ 1, 2 ] ],
  autGroup := Group( [ (2,3), (1,2) ] ),
  group := Group( [ () ] ),
  isGraph := true,
  isSimple := true,
  order := 3,
  representatives := [ 1, 2, 3 ],
  schreierVector := [ -1, -2, -3 ] )
gap>
```

GRAPE – povezivanje s vanjskim programima

GRAPE includes B. D. McKay's nauty (version 2.2) package for calculating automorphism groups of graphs and for testing graph isomorphism. As described in Section *Installing the GRAPE Package*, a user may instead use their own copy of dreadnaut or dreadnautB from a later version of nauty, or may use T. Juntilla's and P. Kaski's bliss package instead of nauty.

GRAPE – povezivanje s vanjskim programima

GRAPE includes B. D. McKay's nauty (version 2.2) package for calculating automorphism groups of graphs and for testing graph isomorphism. As described in Section *Installing the GRAPE Package*, a user may instead use their own copy of dreadnaut or dreadnautB from a later version of nauty, or may use T. Juntilla's and P. Kaski's bliss package instead of nauty.

Except for the nauty package of Brendan D. McKay, the function *SmallestImageSet* by Steve Linton, the **nauty interface** by Alexander Hulpke, and the initial **bliss interface** by Jerry James, the GRAPE package was designed and written by Leonard H. Soicher.

GRAPE – povezivanje s vanjskim programima

GRAPE includes B. D. McKay's nauty (version 2.2) package for calculating automorphism groups of graphs and for testing graph isomorphism. As described in Section *Installing the GRAPE Package*, a user may instead use their own copy of dreadnaut or dreadnautB from a later version of nauty, or may use T. Juntilla's and P. Kaski's bliss package instead of nauty.

Except for the nauty package of Brendan D. McKay, the function *SmallestImageSet* by Steve Linton, the **nauty interface** by Alexander Hulpke, and the initial **bliss interface** by Jerry James, the GRAPE package was designed and written by Leonard H. Soicher.

GRAPE za računanje pune grupe automorfizama i provjeru izomorfnosti grafova koristi dreadnaut, tekstualno sučelje za nauty. Komunikacija s vanjskim programom obavlja se preko tekstualnih datoteka. Kako se izbjegne **konflikt imena datoteka** kad se paralelno izvodi više kopija GAP-a?

DirectoryTemporary

`DirectoryTemporary( )` returns a directory object in the category `IsDirectory` for a new temporary directory. This is guaranteed to be newly created and empty immediately after the call to `DirectoryTemporary`. GAP will make a reasonable effort to remove this directory upon termination of the GAP job that created the directory. If `DirectoryTemporary` is unable to create a new directory, fail is returned. In this case `LastSystemError` can be used to get information about the error. A warning message is given if more than 1000 temporary directories are created in any GAP session.

DirectoryTemporary

`DirectoryTemporary( )` returns a directory object in the category `IsDirectory` for a new temporary directory. This is guaranteed to be newly created and empty immediately after the call to `DirectoryTemporary`. GAP will make a reasonable effort to remove this directory upon termination of the GAP job that created the directory. If `DirectoryTemporary` is unable to create a new directory, fail is returned. In this case `LastSystemError` can be used to get information about the error. A warning message is given if more than 1000 temporary directories are created in any GAP session.

```
gap> GRAPE_nautytmpdir;
dir("c:/WINDOWS/Temp/tm8XUiQa/")
```

GRAPE – povezivanje s vanjskim programima

```
BindGlobal("AutGroupGraph",function(arg)
#
# Let  gr:=arg[1]  be a graph or a graph with colour-classes.
#
# If arg[2] is unbound (the ususal case) then this function re
# the automorphism group of  gr  (making use of B.McKay's
# dreadnaut, nauty  programs).
#
# If arg[2] is bound then  gr  must be a graph and arg[2] is
# a vertex-colouring (not necessarily proper) for  gr
# (i.e. a list of colour-classes for the vertices of gr),
# in which case the subgroup of Aut(gr) preserving this colour
# is returned instead of the full automorphism group.
```

GRAPE – povezivanje s vanjskim programima

```
BindGlobal("GraphIsomorphism",function(arg)
#
# Let  gr1:=arg[1]  and  gr2:=arg[2]  both be graphs or both b
# graphs with colour-classes.
# Then this function returns an isomorphism from  gr1  to  gr2
# gr1  and  gr2  are isomorphic,  else returns  fail.
#

BindGlobal("IsIsomorphicGraph",function(arg)
#
# Let  gr1:=arg[1]  and  gr2:=arg[2]  both be graphs or both b
# graphs with colour-classes.
# Then this function returns true if  gr1  and  gr2  are isomo
# else returns  false.
#
```

GRAPE – povezivanje s vanjskim programima

```
BindGlobal("GraphIsomorphismClassRepresentatives",function(arg
#
# Given a list  L:=arg[1]  of graphs, or of graphs with colour
# this function returns a list
# containing pairwise non-isomorphic elements of  L,  represen
# all the isomorphism classes of elements of  L.
#
```

GRAPE – povezivanje s vanjskim programima

```
BindGlobal("GraphIsomorphismClassRepresentatives",function(arg
#
# Given a list  L:=arg[1]  of graphs, or of graphs with colour
# this function returns a list
# containing pairwise non-isomorphic elements of  L,  represen
# all the isomorphism classes of elements of  L.
#
BindGlobal("ReadOutputNauty",function(file)
# by Alexander Hulpke
  local f, bas, sgens, l, s, p, i, deg, processperm, pi;

  processperm:=function()
    if Length(pi)=0 then return; fi;
    if deg=fail then
      deg:=Length(pi);
    else
```

GRAPE – povezivanje s vanjskim programima

```
BindGlobal("ReadCanonNauty",function(file)
# by Alexander Hulpke
  local f, can, l, deg, s, i;
  f:=InputTextFile(file);
  if f=fail then
    Error("cannot find canonization produced by ‘dreadnaut’");
  fi;


BindGlobal("PrintStreamNautyGraph",function(stream,gamma,col)
  local adj, i, j;
  PrintTo(stream,"d\n$1n",gamma.order,"g\n");
  for i in [1..gamma.order] do
    adj:=Adjacency(gamma,i);
    if adj=[] then
      if i<gamma.order then
AppendTo(stream,";\n");
```

Paket NautyTracesInterface

https://gap-packages.github.io/NautyTracesInterface/

NautyTracesInterface

An interface to nauty

Version 0.2
Released 2018-03-01

Download .tar.gz

View On GitHub

This project is maintained by
The GAP Team

GAP Package NautyTracesInterface

The current version of this package is version 0.2, released on 2018-03-01. For more information, please refer to the package manual. There is also a README file.

Dependencies

This package requires GAP version >= 4.11

The following other GAP packages are needed:

- GAPDoc >= 1.5

Author

Sebastian Gutsche.

Paket NautyTracesInterface

- Ne koristi dreadnaut, nego vlastito sučelje prema nautyju pisano u C-u. Komunikacija između sučelja i GAP-a? (Tekstualne datoteke?)

Paket NautyTracesInterface

- Ne koristi dreadnaut, nego vlastito sučelje prema nautyju pisano u C-u. Komunikacija između sučelja i GAP-a? (Tekstualne datoteke?)
- U GAP-u koristi vlastitu strukturu podataka `NautyGraph`.

Paket NautyTracesInterface

- Ne koristi dreadnaut, nego vlastito sučelje prema nautyju pisano u C-u. Komunikacija između sučelja i GAP-a? (Tekstualne datoteke?)
- U GAP-u koristi vlastitu strukturu podataka `NautyGraph`.
- Paket nije distribuiran s GAP-om, a link "Download .tar.gz" ne radi.

Paket NautyTracesInterface

- Ne koristi dreadnaut, nego vlastito sučelje prema nautyju pisano u C-u. Komunikacija između sučelja i GAP-a? (Tekstualne datoteke?)
- U GAP-u koristi vlastitu strukturu podataka `NautyGraph`.
- Paket nije distribuiran s GAP-om, a link "Download .tar.gz" ne radi.
- Jedini GAP paket koji koristi NautyTracesInterface je Digraphs. Digraphs "po defaultu" koristi bliss, a ako je instaliran ovaj paket može i nauty.

Paket NautyTracesInterface

- Ne koristi dreadnaut, nego vlastito sučelje prema nautyju pisano u C-u. Komunikacija između sučelja i GAP-a? (Tekstualne datoteke?)
- U GAP-u koristi vlastitu strukturu podataka `NautyGraph`.
- Paket nije distribuiran s GAP-om, a link "Download .tar.gz" ne radi.
- Jedini GAP paket koji koristi NautyTracesInterface je Digraphs. Digraphs "po defaultu" koristi bliss, a ako je instaliran ovaj paket može i nauty.
- Kako Digraphs komunicira bliss-om?

Digraphs – povezivanje s vanjskim programima

```
# Wrappers for the C-level functions

BindGlobal("BLISS_DATA_NC",
function(digraph, vert_colours, edge_colours)
  local collapsed, mults, data, edge_gp;
  if IsMultiDigraph(digraph) then
    collapsed      := DIGRAPHS_CollapseMultiColouredEdges(digra
    digraph        := collapsed[1];
    edge_colours   := collapsed[2];
    mults          := collapsed[3];

    data := DIGRAPH_AUTOMORPHISMS(digraph,
                                  vert_colours,
                                  edge_colours);
    if IsEmpty(data[1]) then
      data[1] := [()];
    fi;
```

DESIGN – struktura podataka za dizajne

DESIGN is a package for constructing, classifying, partitioning, and studying block designs. The DESIGN package is written entirely in GAP code, and requires no further installation. However, DESIGN makes use of the GRAPE package, which must be fully installed.

DESIGN – struktura podataka za dizajne

DESIGN is a package for constructing, classifying, partitioning, and studying block designs. The DESIGN package is written entirely in GAP code, and requires no further installation. However, DESIGN makes use of the GRAPE package, which must be fully installed.

The structure of a block design in DESIGN

A **block design** is an ordered pair (X, B), where X is a non-empty finite set whose elements are called points, and B is a non-empty finite multiset whose elements are called blocks, such that each block is a non-empty finite multiset of points.

DESIGN – struktura podataka za dizajne

DESIGN is a package for constructing, classifying, partitioning, and studying block designs. The DESIGN package is written entirely in GAP code, and requires no further installation. However, DESIGN makes use of the GRAPE package, which must be fully installed.

The structure of a block design in DESIGN

A **block design** is an ordered pair (X, B), where X is a non-empty finite set whose elements are called points, and B is a non-empty finite multiset whose elements are called blocks, such that each block is a non-empty finite multiset of points.

DESIGN deals with arbitrary block designs. However, at present, some functions only work for **binary** block designs (i.e. those with no repeated element in any block of the design), but these functions will check if an input block design is binary.

DESIGN – struktura podataka za dizajne

In DESIGN, a block design D is stored as a record, with mandatory components `isBlockDesign`, `v`, and `blocks`. The points of a block design D are always $1, 2, \ldots, D.\mathtt{v}$, but they may also be given names in the optional component `pointNames`, with $D.\mathtt{pointNames}[i]$ the name of point i. The blocks component must be a sorted list of the blocks of D (including any repeats), with each block being a sorted list of points (including any repeats).

DESIGN – struktura podataka za dizajne

In DESIGN, a block design D is stored as a record, with mandatory components `isBlockDesign`, `v`, and `blocks`. The points of a block design D are always $1, 2, \ldots, D.\mathtt{v}$, but they may also be given names in the optional component `pointNames`, with $D.\mathtt{pointNames}[i]$ the name of point i. The blocks component must be a sorted list of the blocks of D (including any repeats), with each block being a sorted list of points (including any repeats).

A block design record may also have some optional components which store information about the design. At present these optional components include `isSimple`, `isBinary`, `isConnected`, `r`, `blockSizes`, `blockNumbers`, `resolutions`, `autGroup`, `autSubgroup`, `tSubsetStructure`, `allTDesignLambdas`, `efficiency`, `id`, `statistical_propertiesXML`, and `pointNames`.

DESIGN – funkcije za kreiranje dizajna

Functions to construct block designs

- `BlockDesign`(v, B)
- `BlockDesign`(v, B, G)

Let v be a positive integer and B a non-empty list of non-empty sorted lists of elements of $\{1, \ldots, v\}$.

The first version of this function returns the block design with point-set $\{1, \ldots, v\}$ and block multiset C, where C is `SortedList`(B).

For the second version of this function, we require G to be a group of permutations of $\{1, \ldots, v\}$, and the function returns the block design with point-set $\{1, \ldots, v\}$ and block multiset C, where C is the sorted list of the concatenation of the G-orbits of the elements of B.

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Display(Fano);
rec(
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBlockDesign := true,
  v := 7 )
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Display(Fano);
rec(
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBlockDesign := true,
  v := 7 )
gap> AllTDesignLambdas(Fano);
[ 7, 3, 1 ]
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
            CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
            Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
            Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
            Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
     [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Display(Fano);
rec(
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBlockDesign := true,
  v := 7 )
gap> AllTDesignLambdas(Fano);
[ 7, 3, 1 ]
gap> Display(Fano);
rec(
  allTDesignLambdas := [ 7, 3, 1 ],
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blockSizes := [ 3 ],
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBinary := true,
  isBlockDesign := true,
  v := 7 )
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
            Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
            Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
     [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Display(Fano);
rec(
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBlockDesign := true,
  v := 7 )
gap> AllTDesignLambdas(Fano);
[ 7, 3, 1 ]
gap> Display(Fano);
rec(
  allTDesignLambdas := [ 7, 3, 1 ],
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blockSizes := [ 3 ],
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBinary := true,
  isBlockDesign := true,
  v := 7 )
gap> AutomorphismGroup(Fano);
Group([ (1,2)(5,7), (2,3)(4,7), (3,5)(6,7), (3,6)(5,7) ])
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Display(Fano);
rec(
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBlockDesign := true,
  v := 7 )
gap> AllTDesignLambdas(Fano);
[ 7, 3, 1 ]
gap> Display(Fano);
rec(
  allTDesignLambdas := [ 7, 3, 1 ],
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blockSizes := [ 3 ],
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBinary := true,
  isBlockDesign := true,
  v := 7 )
gap> AutomorphismGroup(Fano);
Group([ (1,2)(5,7), (2,3)(4,7), (3,5)(6,7), (3,6)(5,7) ])
gap> StructureDescription(last);
"PSL(3,2)"
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
     [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Display(Fano);
rec(
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBlockDesign := true,
  v := 7 )
gap> AllTDesignLambdas(Fano);
[ 7, 3, 1 ]
gap> Display(Fano);
rec(
  allTDesignLambdas := [ 7, 3, 1 ],
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blockSizes := [ 3 ],
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBinary := true,
  isBlockDesign := true,
  v := 7 )
gap> AutomorphismGroup(Fano);
Group([ (1,2)(5,7), (2,3)(4,7), (3,5)(6,7), (3,6)(5,7) ])
gap> StructureDescription(last);
"PSL(3,2)"
gap> Display(Fano);
rec(
  allTDesignLambdas := [ 7, 3, 1 ],
  autGroup := Group( [ (1,2)(5,7), (2,3)(4,7), (3,5)(6,7), (3,6)(5,7) ] ),
  autSubgroup := Group( [ (1,2,3,4,5,6,7) ] ),
  blockSizes := [ 3 ],
  blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
  isBinary := true,
  isBlockDesign := true,
  isSimple := true,
  v := 7 )
gap>
```

DESIGN – funkcije za kreiranje dizajna

- `AGPointFlatBlockDesign`(n, q, d)

Let n be a positive integer, q a prime-power, and d a non-negative integer less than or equal to n. Then this function returns the block design whose points are the points of the affine space $AG(n, q)$, and whose blocks are the d-flats of $AG(n, q)$, considering a d-flat as a set of points.

DESIGN – funkcije za kreiranje dizajna

- `AGPointFlatBlockDesign`(n, q, d)

Let n be a positive integer, q a prime-power, and d a non-negative integer less than or equal to n. Then this function returns the block design whose points are the points of the affine space $AG(n, q)$, and whose blocks are the d-flats of $AG(n, q)$, considering a d-flat as a set of points.

- `PGPointFlatBlockDesign`(n, q, d)

Let n be a non-negative integer, q a prime-power, and d a non-negative integer less than or equal to n. Then this function returns the block design whose points are the (projective) points of the projective space $PG(n, q)$, and whose blocks are the d-flats of $PG(n, q)$, considering a d-flat as a set of projective points.

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0

    GAP      GAP 4.11.0 of 29-Feb-2020
             https://www.gap-system.org
             Architecture: x86_64-pc-cygwin-default64-kv7
 Configuration:  gmp 6.1.2, GASMAN, readline
 Loading the library and packages ...
 Packages:   AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2, Browse 1.8.8,
             CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTblLib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
             Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
             Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
             Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Fano2:=PGPointFlatBlockDesign(2,2,1);
rec( autSubgroup := Group([ (4,6)(5,7), (1,2,4)(3,6,5) ]),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)> ], v := 7 )
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
            Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9,
            Polycyclic 2.15.1, PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
            Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Fano2:=PGPointFlatBlockDesign(2,2,1);
rec( autSubgroup := Group([ (4,6)(5,7), (1,2,4)(3,6,5) ]),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)> ], v := 7 )
gap> Display(Fano2);
rec(
  autSubgroup := Group( [ (4,6)(5,7), (1,2,4)(3,6,5) ] ),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ VectorSpace( GF(2), [ [ 0*Z(2), 0*Z(2), Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ 0*Z(2), Z(2)^0, 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ 0*Z(2), Z(2)^0, Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ Z(2)^0, 0*Z(2), 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ Z(2)^0, 0*Z(2), Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ Z(2)^0, Z(2)^0, 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ Z(2)^0, Z(2)^0, Z(2)^0 ] ] ) ],
  v := 7 )
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
            Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Fano2:=PGPointFlatBlockDesign(2,2,1);
rec( autSubgroup := Group([ (4,6)(5,7), (1,2,4)(3,6,5) ]),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)> ], v := 7 )
gap> Display(Fano2);
rec(
  autSubgroup := Group( [ (4,6)(5,7), (1,2,4)(3,6,5) ] ),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ VectorSpace( GF(2), [ [ 0*Z(2), 0*Z(2), Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ 0*Z(2), Z(2)^0, 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ 0*Z(2), Z(2)^0, Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ Z(2)^0, 0*Z(2), 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ Z(2)^0, 0*Z(2), Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ Z(2)^0, Z(2)^0, 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ Z(2)^0, Z(2)^0, Z(2)^0 ] ] ) ],
  v := 7 )
gap> AllTDesignLambdas(Fano2);
[ 7, 3, 1 ]
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
gap> LoadPackage("DESIGN");
-------------------------------------------------------------------------------
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Fano2:=PGPointFlatBlockDesign(2,2,1);
rec( autSubgroup := Group([ (4,6)(5,7), (1,2,4)(3,6,5) ]),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)> ], v := 7 )
gap> Display(Fano2);
rec(
  autSubgroup := Group( [ (4,6)(5,7), (1,2,4)(3,6,5) ] ),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ VectorSpace( GF(2), [ [ 0*Z(2), 0*Z(2), Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ 0*Z(2), Z(2)^0, 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ 0*Z(2), Z(2)^0, Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ Z(2)^0, 0*Z(2), 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ Z(2)^0, 0*Z(2), Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ Z(2)^0, Z(2)^0, 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ Z(2)^0, Z(2)^0, Z(2)^0 ] ] ) ],
  v := 7 )
gap> AllTDesignLambdas(Fano2);
[ 7, 3, 1 ]
gap> IsIsomorphicBlockDesign(Fano,Fano2);
true
gap>
```

DESIGN – funkcije za kreiranje dizajna

```
/proc/cygdrive/C/gap-4.11.0/gap.exe -l /proc/cygdrive/C/gap-4.11.0
Report issues at https://github.com/gap-packages/grape/issues
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
-------------------------------------------------------------------------------
true
gap> Fano:=BlockDesign(7,[[1,2,4]],CyclicGroup(IsPermGroup,7));
rec( autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ],
      [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ], isBlockDesign := true, v := 7 )
gap> Fano2:=PGPointFlatBlockDesign(2,2,1);
rec( autSubgroup := Group([ (4,6)(5,7), (1,2,4)(3,6,5) ]),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)>, <vector space of dimension 1 over GF(2)>,
      <vector space of dimension 1 over GF(2)> ], v := 7 )
gap> Display(Fano2);
rec(
  autSubgroup := Group( [ (4,6)(5,7), (1,2,4)(3,6,5) ] ),
  blocks := [ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 2, 5, 7 ], [ 3, 4, 7 ], [ 3, 5, 6 ] ],
  isBlockDesign := true,
  pointNames := [ VectorSpace( GF(2), [ [ 0*Z(2), 0*Z(2), Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ 0*Z(2), Z(2)^0, 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ 0*Z(2), Z(2)^0, Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ Z(2)^0, 0*Z(2), 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ Z(2)^0, 0*Z(2), Z(2)^0 ] ] ),
      VectorSpace( GF(2), [ [ Z(2)^0, Z(2)^0, 0*Z(2) ] ] ), VectorSpace( GF(2), [ [ Z(2)^0, Z(2)^0, Z(2)^0 ] ] ) ],
  v := 7 )
gap> AllTDesignLambdas(Fano2);
[ 7, 3, 1 ]
gap> IsIsomorphicBlockDesign(Fano,Fano2);
true
gap> BlockDesignIsomorphismClassRepresentatives([Fano,Fano2]);
[ rec( autGroup := Group([ (1,2)(5,7), (2,4)(5,6), (2,3)(4,7), (3,5)(6,7), (3,7)(5,6) ]),
      autSubgroup := Group([ (1,2,3,4,5,6,7) ]), blockNumbers := [ 7 ], blockSizes := [ 3 ],
      blocks := [ [ 1, 2, 4 ], [ 1, 3, 7 ], [ 1, 5, 6 ], [ 2, 3, 5 ], [ 2, 6, 7 ], [ 3, 4, 6 ], [ 4, 5, 7 ] ],
      isBinary := true, isBlockDesign := true, v := 7 ) ]
gap>
```

Paket IncidenceStructures

https://nagygp.github.io/IncidenceStructures/

IncidenceStructures

GAP implementation of abstract incidence structures

Version 0.3
Released 2020-11-05

Download .tar.gz

View On GitHub

This project is maintained by Gábor P. Nagy

GAP Package IncidenceStructures

The current version of this package is version 0.3, released on 2020-11-05. For more information, please refer to the package manual. There is also a README file.

Dependencies

This package requires GAP version >= 4.9

The following other GAP packages are needed:

- BlissInterface >=0.2

Author

Gábor P. Nagy.

Paket IncidenceStructures

- `IncidenceStructureByIncidenceMatrix`($pts, bls, incmat$)
- `IncidenceStructureByBlocks`(pts, bls)
- `AutomorphismGroup`(s)
- `Isomorphism`(s_1, s_2)

Canonocal labellings, automorphisms and isomorphisms of incidence structures are computed with the help of their incidence graphs, which are bipartite graphs with upper vertices the points and lower vertices the blocks. The computation is done using the GAP package **BlissInterface**.

Paket BlissInterface

https://gap-packages.github.io/BlissInterface/

BlissInterface

Low level interface to the bliss graph automorphism tool

Version 0.22
Released 2020-03-26

Download .tar.gz

View On GitHub

This project is maintained by Gábor P. Nagy

GAP Package BlissInterface

The current version of this package is version 0.22, released on 2020-03-26. For more information, please refer to the package manual. There is also a README file.

Dependencies

This package requires GAP version >= 4.9

Author

Gábor P. Nagy.

Paket BlissInterface

```
/*
 * BlissInterface: Low level interface to the bliss graph auto
 */

#include "../extern/bliss-0.73/graph.hh" /* for bliss graph cl
#include "compiled.h" // GAP headers

/*
 * The following code is a derivative work of the code from th
 * Digraphs which is licensed GPLv3. This code therefore is al
 * the terms of the GNU Public License, version 3.
 *
 * Based on "digraphs_hook_function()" of the GAP package Digr
 * Modified by G.P. Nagy, 21/08/2019
 */

void blissinterface_hook_function(void *user_param_v, unsigned
```

Kraj

Hvala na *e*-pažnji!