GAP - Changes from Earlier Versions

Release 4.7.9, 29-Nov-2015

The GAP Group

The GAP Group Email: support@gap-system.org
Homepage: http://www.gap-system.org

Copyright

Copyright © (1987-2015) for the core part of the GAP system by the GAP Group.

Most parts of this distribution, including the core part of the GAP system are distributed under the terms of the GNU General Public License, see http://www.gnu.org/licenses/gpl.html or the file GPL in the etc directory of the GAP installation.

More detailed information about copyright and licenses of parts of this distribution can be found in Section (**Reference: Copyright and License**) of the GAP reference manual.

GAP is developed over a long time and has many authors and contributors. More detailed information can be found in Section (**Reference: Authors and Maintainers**) of the GAP reference manual.

Contents

1 Preface **5**

2 Changes between GAP 4.6 and GAP 4.7 **6**
2.1 GAP 4.7.2 (December 2013) . . . 6
2.2 GAP 4.7.3 (February 2014) . . . 11
2.3 GAP 4.7.4 (February 2014) . . . 11
2.4 GAP 4.7.5 (May 2014) . . . 11
2.5 GAP 4.7.6 (November 2014) . . . 12
2.6 GAP 4.7.7 (February 2015) . . . 13
2.7 GAP 4.7.8 (June 2015) . . . 14
2.8 GAP 4.7.9 (November 2015) . . . 15

3 Changes between GAP 4.5 and GAP 4.6 **16**
3.1 GAP 4.6.2 (February 2013) . . . 16
3.2 GAP 4.6.3 (March 2013) . . . 18
3.3 GAP 4.6.4 (April 2013) . . . 20
3.4 GAP 4.6.5 (July 2013) . . . 21

4 Changes between GAP 4.4 and GAP 4.5 **22**
4.1 Changes in the core GAP system introduced in GAP 4.5 . . . 22
4.2 Packages in GAP 4.5 . . . 29
4.3 GAP 4.5.5 (July 2012) . . . 34
4.4 GAP 4.5.6 (September 2012) . . . 35
4.5 GAP 4.5.7 (December 2012) . . . 37

5 Overview of updates for GAP 4.4 **39**
5.1 GAP 4.4 Bugfix 2 (April 2004) . . . 39
5.2 GAP 4.4 Bugfix 3 (May 2004) . . . 39
5.3 GAP 4.4 Bugfix 4 (December 2004) . . . 40
5.4 GAP 4.4 Update 5 (May 2005) . . . 41
5.5 GAP 4.4 Update 6 (September 2005) . . . 45
5.6 GAP 4.4 Update 7 (March 2006) . . . 49
5.7 GAP 4.4 Update 8 (September 2006) . . . 52
5.8 GAP 4.4 Update 9 (November 2006) . . . 55
5.9 GAP 4.4 Update 10 (October 2007) . . . 55
5.10 GAP 4.4 Update 11 (December 2008) . . . 58
5.11 GAP 4.4 Update 12 (December 2008) . . . 61

6 Changes from Earlier Versions **62**
6.1 Changes between GAP 4.3 and GAP 4.4 62
6.2 Earlier Changes 65

Index **68**

Chapter 1

Preface

This is one of the three main GAP books. It describes most essential changes from previous GAP releases.

In addition to this manual, there is the *GAP Tutorial* and the *GAP Reference Manual* containing detailed documentation of the mathematical functionality of GAP.

A lot of the functionality of the system and a number of contributed extensions are provided as GAP packages, and each of these has its own manual. New versions of packages are released independently of GAP releases, and changes between package versions may be described in their documentation.

Chapter 2

Changes between GAP 4.6 and GAP 4.7

This chapter contains an overview of most important changes introduced in GAP 4.7.2 release (the first public release of GAP 4.7). It also contains information about subsequent update releases for GAP 4.7.

2.1 GAP 4.7.2 (December 2013)

2.1.1 Changes in the core GAP system introduced in GAP 4.7

Improved and extended functionality:

- The methods for computing conjugacy classes of permutation groups have been rewritten from scratch to enable potential use for groups in different representations. As a byproduct the resulting code is (sometimes notably) faster. It also now is possible to calculate canonical conjugacy class representatives in permutation groups, which can be beneficial when calculating character tables.
- The methods for determining (conjugacy classes of) subgroups in non-solvable groups have been substantially improved in speed and scope for groups with multiple nonabelian composition factors.
- There is a new method for calculating the maximal subgroups of a permutation group (with chief factors of width less or equal 5) without calculating the whole subgroup lattice.
- If available, information from the table of marks library is used to speed up subgroup calculations in almost simple factor groups.
- The broader availability of maximal subgroups is used to improve the calculation of double cosets.
- To illustrate the improvements listed above, one could try, for example

Example

```
g:=WreathProduct(MathieuGroup(11),Group((1,2)));
Length(ConjugacyClassesSubgroups(g));
```

and

Example

```
g:=SemidirectProduct(GL(3,5),GF(5)^3);
g:=Image(IsomorphismPermGroup(g));
MaximalSubgroupClassReps(g);
```

- Computing the exponent of a finite group G could be extremely slow. This was due to a slow default method being used, which computed all conjugacy classes of elements in order to compute the exponent. We now instead compute Sylow subgroups P_1, ..., P_k of G and use the easily verified equality $exp(G) = exp(P_1)x...xexp(P_k)$. This is usually at least as fast and in many cases orders of magnitude faster.

Example

```
gap> G:=SmallGroup(2^7*9,33);;
gap> H:=DirectProduct(G, ElementaryAbelianGroup(2^10));;
gap> Exponent(H); # should take at most a few milliseconds
72
gap> K := PerfectGroup(2688,3);;
gap> Exponent(K); # should take at most a few seconds
168
```

- The functionality in GAP for transformations and transformation semigroups has been rewritten and extended. Partial permutations and inverse semigroups have been newly implemented. The documentation for transformations and transformation semigroups has been improved. Transformations and partial permutations are implemented in the GAP kernel. Methods for calculating attributes of transformations and partial permutations, and taking products, and so are also implemented in the kernel. The new implementations are largely backwards compatible; some exceptions are given below.

  The degree of a transformation `f` is usually defined as the largest positive integer where `f` is defined. In previous versions of GAP, transformations were only defined on positive integers less than their degree, it was only possible to multiply transformations of equal degree, and a transformation did not act on any point exceeding its degree. Starting with GAP 4.7, transformations behave more like permutations, in that they fix unspecified points and it is possible to multiply arbitrary transformations.

  – in the display of a transformation, the trailing fixed points are no longer printed. More precisely, in the display of a transformation `f` if `n` is the largest value such that `n^f<>n` or `i^f=n` for some `i<>n`, then the values exceeding `n` are not printed.
  – the display for semigroups of transformations now includes more information, for example `<transformation semigroup on 10 pts with 10 generators>` and `<inverse partial perm semigroup on 10 pts with 10 generators>`.
  – transformations which define a permutation can be inverted, and groups of transformations can be created.

  Further information regarding transformations and partial permutations, can be found in the relevant chapters of the reference manual.

  The code for Rees matrix semigroups has been completely rewritten to fix the numerous bugs in the previous versions. The display of a Rees matrix semigroup has also been improved to include the numbers of rows and columns, and the underlying semigroup. Again the new implementations should be backwards compatible with the exception that the display is different.

The code for magmas with a zero adjoined has been improved so that it is possible to access more information about the original magma. The display has also been changed to indicate that the created magma is a magma with zero adjoined (incorporating the display of the underlying magma). Elements of a magma with zero are also printed so that it is clear that they belong to a magma with zero.

If a semigroup is created by generators in the category IsMultiplicativeElementWithOneCollection and CanEasilyCompareElements, then it is now checked if the One of the generators is given as a generator. In this case, the semigroup is created as a monoid.

- Added a new operation `GrowthFunctionOfGroup` (**Reference: GrowthFunctionOfGroup**) that gives sizes of distance spheres in the Cayley graph of a group.
- A new group constructor `FreeAbelianGroup` (**Reference: FreeAbelianGroup**) for free abelian groups has been added. By default, it creates suitable fp groups. Though free abelian groups groups do not offer much functionality right now, in the future other implementations may be provided, e.g. by the Polycyclic package.
- The message about halving the pool size at startup is only shown when `-D` command line option is used (see (**Reference: Command Line Options**)). [Suggested by Volker Braun]
- An info class called `InfoObsolete` (**Reference: InfoObsolete**) with the default level 0 is introduced. Setting it to 1 will trigger warnings at runtime if an obsolete variable declared with `DeclareObsoleteSynonym` is used. This is recommended for testing GAP distribution and packages.
- The GAP help system now recognises some common different spelling patterns (for example, -ise/-ize, -isation/-ization, solvable/soluble) and searches for all possible spelling options even when the synonyms are not declared.
- Added new function `Cite` (**Reference: Cite**) which produces citation samples for GAP and packages.
- It is now possible to compile GAP with user-supplied `CFLAGS` which now will not be overwritten by GAP default settings. [Suggested by Jeroen Demeyer]

Fixed bugs:

- `Union` (**Reference: Union**) had $O(n^3)$ behaviour when given many ranges (e.g. it could take 10 seconds to find a union of 1000 1-element sets). The new implementation reduces that to $O(nlogn)$ (and 4ms for the 10 second example), at the cost of not merging ranges as well as before in some rare cases.
- `IsLatticeOrderBinaryRelation` only checked the existence of upper bounds but not the uniqueness of the least upper bound (and dually for lower bounds), so in some cases it could return the wrong answer. [Reported by Attila Egri-Nagy]
- `LowIndexSubgroupsFpGroup` (**Reference: LowIndexSubgroupsFpGroup**) triggered a break loop if the list of generators of the 2nd argument contained the identity element of the group. [Reported by Ignat Soroko]

- Fixed regression in heuristics used by NaturalHomomorphismByNormalSubgroup (**Reference: NaturalHomomorphismByNormalSubgroup**) that could produce a permutation representation of an unreasonably large degree. [Reported by Izumi Miyamoto]

- Fixed inconsistent behaviour of QuotientMod(Integers, r, s, m) in the case where s and m are not coprime. This fix also corrects the division behaviour of ZmodnZ objects, see QuotientMod (**Reference: QuotientMod**) and ZmodnZ (**Reference: ZmodnZ**). [Reported by Mark Dickinson]

- Fixed an oversight in the loading process causing OnQuit (**Reference: OnQuit**) not resetting the options stack after exiting the break loop.

- Empty strings were treated slightly differently than other strings in the GAP kernel, for historical reasons. This resulted in various inconsistencies. For example, IsStringRep("") returned true, but a method installed for arguments of type IsStringRep would NOT be invoked when called with an empty string.

  We remove this special case in the GAP kernel (which dates back the very early days of GAP 4 in 1996). This uncovered one issue in the kernel function POSITION_SUBSTRING (when calling it with an empty string as second argument), which was also fixed.

- The parser for floating point numbers contained a bug that could cause GAP to crash or to get into a state where the only action left to the user was to exit GAP via Ctrl-D. For example, entering four dots with spaces between them on the GAP prompt and then pressing the return key caused GAP to exit.

  The reason was (ironically) an error check in the innards of the float parser code which invoked the GAP Error() function at a point where it should not have.

- Removing the last character in a string was supposed to overwrite the old removed character in memory with a zero byte, but failed to do so due to an off-by-one error. For most GAP operations, this has no visible effect, except for those which directly operate on the underlying memory representation of strings. For example, when trying to use such a string to reference a record entry, a (strange) error could be triggered.

- ViewString (**Reference: ViewString**) and DisplayString (**Reference: DisplayString**) are now handling strings, characters and immediate FFEs in a consistent manner.

- Multiple fixes to the build process for less common Debian platforms (arm, ia64, mips, sparc, GNU/Hurd). [Suggested by Bill Allombert]

- Fixes for several regressions in the gac script. [Suggested by Bill Allombert]

Changed functionality:

- It is not possible now to call WreathProduct (**Reference: WreathProduct**) with 2nd argument H not being a permutation group, without using the 3rd argument specifying the permutation representation. This is an incompatible change but it will produce an error instead of a wrong result. The former behaviour of WreathProduct (**Reference: WreathProduct**) may now be achieved by using StandardWreathProduct (**Reference: StandardWreathProduct**) which returns the wreath product for the (right regular) permutation action of H on its elements.

- The function `ViewLength` to specify the maximal number of lines that are printed in `ViewObj` (**Reference: ViewObj**) became obsolete, since there was already a user preference `ViewLength` to specify this. The value of this preference is also accessible in `GAPInfo.ViewLength`.

2.1.2 New and updated packages since GAP 4.6.5

At the time of the release of GAP 4.6.5 there were 107 packages redistributed with GAP. The first public release of GAP 4.7 contains 114 packages.

One of essential changes is that the Citrus package by J.Mitchell has been renamed to Semigroups. The package has been completely overhauled, the performance has been improved, and the code has been generalized so that in the future the same code can be used to compute with other types of semigroups.

Furthermore, new packages that have been added to the redistribution since the release of GAP 4.6.5 are:

- 4ti2interface package by Sebastian Gutsche, providing an interface to 4ti2, a software package for algebraic, geometric and combinatorial problems on linear spaces (`http://www.4ti2.de`).
- CoReLG by Heiko Dietrich, Paolo Faccin and Willem de Graaf for calculations in real semisimple Lie algebras.
- IntPic package by Manuel Delgado, aimed at providing a simple way of getting a pictorial view of sets of integers. The main goal of the package is producing Tikz code for arrays of integers. The code produced is to be included in a LaTeX file, which can then be processed. Some of the integers are emphasized by using different colors for the cells containing them.
- LieRing by Serena Cicalo and Willem de Graaf for constructing finitely-presented Lie rings and calculating the Lazard correspondence. The package also provides a database of small n-Engel Lie rings.
- LiePRing package by Michael Vaughan-Lee and Bettina Eick, introducing a new datastructure for nilpotent Lie rings of prime-power order. This allows to define such Lie rings for specific primes as well as for symbolic primes and other symbolic parameters. The package also includes a database of nilpotent Lie rings of order at most p^7 for all primes $p > 3$.
- ModIsom by Bettina Eick, which contains various methods for computing with nilpotent associative algebras. In particular, it contains a method to determine the automorphism group and to test isomorphisms of such algebras over finite fields and of modular group algebras of finite p-groups. Further, it contains a nilpotent quotient algorithm for finitely presented associative algebras and a method to determine Kurosh algebras.
- SLA by Willem de Graaf for computations with simple Lie algebras. The main topics of the package are nilpotent orbits, theta-groups and semisimple subalgebras.

Furthermore, some packages have been upgraded substantially since the GAP 4.6.5 release:

- ANUPQ package by Greg Gamble, Werner Nickel and Eamonn O'Brien has been updated after Max Horn joined it as a maintainer. As a result, it is now much easier to install and use it with the current GAP release.

- Wedderga package by Osnel Broche Cristo, Allen Herman, Alexander Konovalov, Aurora Olivieri, Gabriela Olteanu, Ángel del Río and Inneke Van Gelder has been extended to include functions for calculating local and global Schur indices of ordinary irreducible characters of finite groups, cyclotomic algebras over abelian number fields, and rational quaternion algebras (contribution by Allen Herman).

2.2 GAP 4.7.3 (February 2014)

Fixed bugs which could lead to incorrect results:

- Incorrect result returned by `AutomorphismGroup(PSp(4,2^n))`. [Reported by Anvita]
- The `Order` (**Reference: Order**) method for group homomorphisms newly introduced in GAP 4.7 had a bug that caused it to sometimes return incorrect results. [Reported by Benjamin Sambale]

Fixed bugs that could lead to break loops:

- Several bugs were fixed and missing methods were introduced in the new code for transformations, partial permutations and semigroups that was first included in GAP 4.7. Some minor corrections were made in the documentation for transformations.
- Break loop in `IsomorphismFpMonoid` when prefixes in generators names were longer than one letter. [Reported by Dmytro Savchuk and Yevgen Muntyan]
- Break loop while displaying the result of `MagmaWithInversesByMultiplicationTable` (**Reference: MagmaWithInversesByMultiplicationTable**). [Reported by Grahame Erskine]

Improved functionality:

- Better detection of UTF-8 terminal encoding on some systems. [Suggested by Andries Brouwer]

2.3 GAP 4.7.4 (February 2014)

This release was prepared immediately after GAP 4.7.3 to revert the fix of the error handling for the single quote at the end of an input line, contained in GAP 4.7.3. It happened that (only on Windows) the fix caused error messages in one of the packages.

2.4 GAP 4.7.5 (May 2014)

Fixed bugs which could lead to incorrect results:

- `InstallValue` (**Reference: InstallValue**) cannot handle immediate values, characters or booleans for technical reasons. A check for such values was introduced to trigger an error message and prevent incorrect results caused by this. [Reported by Sebastian Gutsche]

- `KnowsDictionary` (**Reference: KnowsDictionary**) and `LookupDictionary` (**Reference: LookupDictionary**) methods for `IsListLookupDictionary` were using `PositionFirstComponent` (**Reference: PositionFirstComponent**); the latter is only valid on sorted lists, but in `IsListLookupDictionary` the underlying list is NOT sorted in general, leading to bogus results.

Other fixed bugs:

- A bug in `DirectProductElementsFamily` which used `CanEasilyCompareElements` (**Reference: CanEasilyCompareElements**) instead of `CanEasilySortElements` (**Reference: CanEasilySortElements**).
- Fixed wrong `Infolevel` message that caused a break loop for some automorphism group computations.
- Fixed an error that sometimes caused a break loop in `HallSubgroup` (**Reference: HallSubgroup**). [Reported by Benjamin Sambale]
- Fixed a rare error in computation of conjugacy classes of a finite group by homomorphic images, providing fallback to a default algorithm.
- Fixed an error in the calculation of Frattini subgroup in the case of the trivial radical.
- Several minor bugs were fixed in the documentation, kernel, and library code for transformations.
- Fixed errors in `NumberPerfectGroups` (**Reference: NumberPerfectGroups**) and `NumberPerfectLibraryGroups` (**Reference: NumberPerfectLibraryGroups**) not being aware that there are no perfect groups of odd order.
- Restored the ability to build GAP on OS X 10.4 and 10.5 which was accidentally broken in the previous GAP release by using the build option not supported by these versions.
- Fixed some problems for ia64 and sparc architectures. [Reported by Bill Allombert and Volker Braun]

New package added for the redistribution with GAP:

- permut package by A.Ballester-Bolinches, E.Cosme-Llópez, and R.Esteban-Romero to deal with permutability in finite groups.

2.5 GAP 4.7.6 (November 2014)

Fixed bugs which could lead to incorrect results:

- A bug that may cause `ShortestVectors` (**Reference: ShortestVectors**) to return an incomplete list. [Reported by Florian Beye]
- A bug that may lead to incorrect results and infinite loops when GAP is compiled without GMP support using gcc 4.9.

- A bug that may cause `OrthogonalEmbeddings` (**Reference: OrthogonalEmbeddings**) to return an incomplete result. [Reported by Benjamin Sambale]

Fixed bugs that could lead to break loops:

- `ClosureGroup` (**Reference: ClosureGroup**) should be used instead of `ClosureSubgroup` (**Reference: ClosureSubgroup**) in case there is no parent group, otherwise some calculations such as e.g. `NormalSubgroups` (**Reference: NormalSubgroups**) may fail. [Reported by Dmitrii Pasechnik]
- Fixed a line in the code that used a hard-coded identity permutation, not a generic identity element of a group. [Reported by Toshio Sumi]
- Fixed a problem in the new code for calculating maximal subgroups that caused a break loop for some groups from the transitive groups library. [Reported by Petr Savicky]
- Fixed a problem in `ClosureSubgroup` (**Reference: ClosureSubgroup**) not accepting some groups without `Parent` (**Reference: Parent**). [Reported by Inneke van Gelder]

Other fixed bugs:

- Eliminated a number of compiler warnings detected with some newer versions of C compilers.
- Some minor bugs in the transformation and partial permutation code and documentation were resolved.

2.6 GAP 4.7.7 (February 2015)

New features:

- Introduced some arithmetic operations for infinity and negative infinity, see **Reference: infinity**.
- Introduced new property `IsGeneratorsOfSemigroup` (**Reference: IsGeneratorsOfSemigroup**) which reflects wheter the list or collection generates a semigroup.

Fixed bugs which could lead to incorrect results:

- Fixed a bug in `Union` (**Reference: Union**) (actually, in the internal library function `JoinRanges`) caused by downward running ranges. [Reported by Matt Fayers]
- Fixed a bug where recursive records might be printed with the wrong component name, coming from component names being ordered differently in two different pieces of code. [Reported by Thomas Breuer]
- The usage of `abs` in `src/gmpints.c` was replaced by `AbsInt`. The former is defined to operate on 32-bit integers even if GAP is compiled in 64-bit mode. That lead to truncating GAP integers and caused a crash in `RemInt` (**Reference: RemInt**), reported by Willem De Graaf and Heiko Dietrich. Using `AbsInt` fixes the crash, and ensures the correct behaviour on 32-bit and 64-bit builds.

Fixed bugs that could lead to break loops:

- A problem with `ProbabilityShapes` (**Reference: ProbabilityShapes**) not setting frequencies list for small degrees. [Reported by Daniel Błażewicz and independently by Mathieu Gagne]
- An error when generating a free monoid of rank infinity. [Reported by Nick Loughlin]
- Several bugs with the code for Rees matrix semigroups not handling trivial cases properly.
- A bug in `IsomorphismTypeInfoFiniteSimpleGroup` (**Reference: IsomorphismTypeInfoFiniteSimpleGroup**) affecting one particular group due to a misformatting in a routine that translates between the Chevalley type and the name used in the table (in this case, `"T"` was used instead of `["T"]`). [Reported by Petr Savicky]

Other fixed bugs:

- The `Basis` (**Reference: Basis**) method for full homomorphism spaces of linear mappings did not set basis vectors which could be obtained by `GeneratorsOfLeftModule` (**Reference: GeneratorsOfLeftModule**).
- A problem with `GaloisType` (**Reference: GaloisType**) entering an infinite loop in the routine for approximating a root. [Reported by Daniel Błażewicz]
- Fixed the crash when GAP is called when the environment variables `HOME` or `PATH` are unset. [Reported by Bill Allombert]

Furthermore, new packages that have been added to the redistribution since the release of GAP 4.7.6 are:

- json package by Christopher Jefferson, providing a mapping between the JSON markup language and GAP
- SglPPow package by Bettina Eick and Michael Vaughan-Lee, providing the database of p-groups of order p^7 for $p > 11$, and of order 3^8.

2.7 GAP 4.7.8 (June 2015)

Fixed bugs which could lead to incorrect results:

- Added two groups of degree 1575 which were missing in the library of first primitive groups. [Reported by Gordon Royle]
- Fixed the error in the code for algebra module elements in packed representation caused by the use of `Objectify` (**Reference: Objectify**) with the type of the given object instead of `ObjByExtRep` (**Reference: ObjByExtRep**) as recommended in (**Reference: Further Improvements in Implementing Residue Class Rings**). The problem was that after calculating `u+v` where one of the summands was known to be zero, this knowledge was wrongly passed to the sum via the type. [Reported by Istvan Szollosi]
- Fixed a bug in `PowerMod` (**Reference: PowerMod**) causing wrong results for univariate Laurent polynomials when the two polynomial arguments are stored with the same non-zero shift. [Reported by Max Horn]

Furthermore, new packages that have been added to the redistribution since the release of GAP 4.7.7 are:

- PatternClass by Michael Albert, Ruth Hoffmann and Steve Linton, allowing to explore the permutation pattern classes build by token passing networks. Amongst other things, it can compute the basis of a permutation pattern class, create automata from token passing networks and check if the deterministic automaton is a possible representative of a token passing network.
- QPA by Edward Green and Øyvind Solberg, providing data structures and algorithms for computations with finite dimensional quotients of path algebras, and with finitely generated modules over such algebras. It implements data structures for quivers, quotients of path algebras, and modules, homomorphisms and complexes of modules over quotients of path algebras.

2.8 GAP 4.7.9 (November 2015)

Fixed bug which could lead to incorrect results:

- Fixed an error in `DataAboutSimpleGroup`calculation, which caused wrong result in calculation of the automorphism group of J_2. [Reported by Fatemeh Koorepazan-Moftakhar]

Improved and extended functionality:

- `GModuleByMats` (**Reference: GModuleByMats**) and some related functions now work for infinite fields, in particular in characteristic 0. However, this only applies for functions that do not need the MeatAxe, as that is still tied to finite fields.

Chapter 3

Changes between GAP 4.5 and GAP 4.6

This chapter lists most important changes between GAP 4.5.7 and GAP 4.6.2 (i.e. between the last release of GAP 4.5 and the first public release of GAP 4.6). It also contains information about subsequent update releases for GAP 4.6.

3.1 GAP 4.6.2 (February 2013)

3.1.1 Changes in the core GAP system introduced in GAP 4.6

Improved and extended functionality:

- It is now possible to declare a name as an operation with two or more arguments (possibly several times) and *THEN* declare it as an attribute. Previously this was only possible in the other order. This should make the system more independent of the order in which packages are loaded.
- Words in fp groups are now printed in factorised form if possible and not too time-consuming, i.e. `a*b*a*b*a*b` will be printed as `(a*b)^3`.
- Added methods to calculate Hall subgroups in nonsolvable groups.
- Added a generic method for `IsPSolvable` (**Reference: IsPSolvable**) and a better generic method for `IsPNilpotent` (**Reference: IsPNilpotent**) for groups.
- Improvements to action homomorphisms: image of an element can use existing stabiliser chain of the image group (to reduce the number of images to compute), preimages under linear/projective action homomorphisms use linear algebra to avoid factorisation.
- To improve efficiency, additional code was added to make sure that the `HomePcgs` of a permutation group is in `IsPcgsPermGroupRep` representation in more cases.
- Added an operation `SortBy` (**Reference: SortBy**) with arguments being a function *`f`* of one argument and a list *`l`* to be sorted in such a way that *`l`*`(`*`f`*`[i]) <= `*`l`*`(`*`f`*`[i+1])`.
- Added a kernel function `MEET_BLIST` which returns `true` if the two boolean lists have `true` in any common position and `false` otherwise. This is useful for representing subsets of a fixed set by boolean lists.

- When assigning to a position in a compressed FFE vector GAP now checks to see if the value being assigned can be converted into an internal FFE element if it isn't one already. This uses new attribute `AsInternalFFE` (**Reference: AsInternalFFE**), for which methods are installed for internal FFEs, Conway FFEs and ZmodpZ objects.
- Replaced `ViewObj` (**Reference: ViewObj**) method for fields by `ViewString` (**Reference: ViewString**) method to improve the way how polynomial rings over algebraic extenstions of fields are displayed.
- Made the info string (optional 2nd argument to `InstallImmediateMethod` (**Reference: InstallImmediateMethod**)) behave similarly to the info string in `InstallMethod` (**Reference: InstallMethod**). In particular, `TraceImmediateMethods` (**Reference: TraceImmediateMethods**) now always prints the name of the operation.
- Syntax errors such as `Unbind(x,1)` had the unhelpful property that `x` got unbound before the syntax error was reported. A specific check was added to catch this and similar cases a little earlier.
- Allow a `GAPARGS` parameter to the top-level GAP `Makefile` to pass extra arguments to the GAP used for manual building.
- Added an attribute `UnderlyingRingElement` (**Reference: UnderlyingRingElement**) for Lie objects.
- The function `PrimeDivisors` (**Reference: PrimeDivisors**) now became an attribute. [suggested by Mohamed Barakat]
- Added an operation `DistancePerms` (**Reference: DistancePerms**) with a kernel method for internal permutations and a generic method.
- Added a method for `Subfields` (**Reference: Subfields**) to support large finite fields. [reported by Inneke van Gelder]

Fixed bugs which could lead to crashes:

- The extremely old `DEBUG_DEADSONS_BAGS` compile-time option has not worked correctly for many years and indeed crashes GAP. The type of bug it is there to detect has not arisen in many years and we have certainly not used this option, so it has been removed. [Reported by Volker Braun]

Other fixed bugs:

- Scanning of floating point literals collided with iterated use of integers as record field elements in expressions like `r.1.2`.
- Fixed two potential problems in `NorSerPermPcgs`, one corrupting some internal data and one possibly mixing up different pcgs.
- Fixed a performance problem with `NiceMonomorphism` (**Reference: NiceMonomorphism**). [reported by John Bamberg]
- Fixed a bug in `ReadCSV` (**Reference: ReadCSV**) that caused some `.csv` files being parsed incorrectly.

No longer supported:

- The file `lib/consistency.g`, which contained three undocumented auxiliary functions, has been removed from the library. In addition, the global record `Revision` is now deprecated, so there is no need to bind its components in GAP packages.

3.1.2 New and updated packages since GAP 4.5.4

At the time of the release of GAP 4.5 there were 99 packages redistributed with GAP. The first public release of GAP 4.6 contains 106 packages.

The new packages that have been added to the redistribution since the release of GAP 4.5.4 are:

- AutoDoc package by S. Gutsche, providing tools for automated generation of GAPDoc manuals.
- Congruence package by A. Konovalov, which provides functions to construct various canonical congruence subgroups in $SL_2(\mathbb{Z})$, and also intersections of a finite number of such subgroups, implements the algorithm for generating Farey symbols for congruence subgroups and uses it to produce a system of independent generators for these subgroups.
- Convex package by S. Gutsche, which provides structures and algorithms for convex geometry.
- Float package by L. Bartholdi, which extends GAP floating-point capabilities by providing new floating-point handlers for high-precision real, interval and complex arithmetic using MPFR, MPFI, MPC or CXSC external libraries. It also contains a very high-performance implementation of the LLL (Lenstra-Lenstra-Lovász) lattice reduction algorithm via the external library FPLLL.
- PolymakeInterface package by T. Baechler and S. Gutsche, providing a link to the callable library of the polymake system (`http://www.polymake.org`).
- ToolsForHomalg package by M. Barakat, S. Gutsche and M. Lange-Hegermann, which provides some auxiliary functionality for the homalg project (`http://homalg.math.rwth-aachen.de/`).
- ToricVarieties package by S. Gutsche, which provides data structures to handle toric varieties by their commutative algebra structure and by their combinatorics.

Furthermore, some packages have been upgraded substantially since the GAP 4.5.4 release:

- Starting from 2.x.x, the functionality for iterated monodromy groups has been moved from the FR package by L. Bartholdi to a separate package IMG (currently undeposited, available from `https://github.com/laurentbartholdi/img`). This completely removes the dependency of FR on external library modules, and should make its installation much easier.

3.2 GAP 4.6.3 (March 2013)

Improved functionality:

- Several changes were made to `IdentityMat` (**Reference: IdentityMat**) and `NullMat` (**Reference: NullMat**). First off, the documentation was changed to properly state that these functions support arbitrary rings, and not just fields. Also, more usage examples were added to the manual.

  For `NullMat`, it is now also always possible to specify a ring element instead of a ring, and this is documented. This matches existing `IdentityMat` behavior, and partially worked before (undocumented), but in some cases could run into error or infinite recursion.

  In the other direction, if a finite field element, `IdentityMat` now really creates a matrix over the smallest field containing that element. Previously, a matrix over the prime field was created instead, contrary to the documentation.

  Furthermore, `IdentityMat` over small finite fields is now substantially faster when creating matrices of large dimension (say a thousand or so).

  Finally, `MutableIdentityMat` (**Reference: MutableIdentityMat**) and `MutableNullMat` (**Reference: MutableNullMat**) were explicitly declared obsolete (and may be removed in GAP 4.7). They actually were deprecated since GAP 4.1, and their use discouraged by the manual. Code using them should switch to `IdentityMat` (**Reference: IdentityMat**) respectively `NullMat` (**Reference: NullMat**).

- Two new `PerfectResiduum` (**Reference: PerfectResiduum**) methods were added for solvable and perfect groups, handling these cases optimally. Moreover, the existing generic method was improved by changing it to use `DerivedSeriesOfGroup` (**Reference: DerivedSeriesOfGroup**). Previously, it would always compute the derived series from scratch and then throw away the result.

- A new `MinimalGeneratingSet` (**Reference: MinimalGeneratingSet**) method for groups handled by a nice monomorphisms was added, similar to the existing `SmallGeneratingSet` (**Reference: SmallGeneratingSet**) method. This is useful if the nice monomorphism is already mapping into a pc or pcp group.

- Added a special method for `DerivedSubgroup` (**Reference: DerivedSubgroup**) if the group is known to be abelian.

Fixed bugs:

- Fixed a bug in `PowerModInt` (**Reference: PowerModInt**) computing r^e mod m in a special case when $e = 0$ and $m = 0$. [Reported by Ignat Soroko]

- `CoefficientsQadic` (**Reference: CoefficientsQadic**) now better checks its arguments to avoid an infinite loop when being asked for a q-adic representation for $q = 1$. [Reported by Ignat Soroko]

- Methods for `SylowSubgroupOp` (see `SylowSubgroup` (**Reference: SylowSubgroup**)) for symmetric and alternating group did not always set `IsPGroup` (**Reference: IsPGroup**) and `PrimePGroup` (**Reference: PrimePGroup**) for the returned Sylow subgroup.

- Display of matrices consisting of Conway field elements (which are displayed as polynomials) did not print constant 1 terms.

- Added an extra check and a better error message in the method to access *natural* generators of domains using the . operator (see `GeneratorsOfDomain` (**Reference: GeneratorsOfDomain**)).
- Trying to solve the word problem in an fp group where one or more generators has a name of more than one alphabetic character led to a break loop.
- Provided the default method for `AbsoluteIrreducibleModules` (**Reference: AbsoluteIrreducibleModules**) as a temporary workaround for the problem which may cause returning wrong results or producing an error when being called for a non-prime field.
- A bug in the GAP kernel caused `RNamObj` to error out when called with a string that had the `IsSSortedList` (**Reference: IsSSortedList**) property set (regardless of whether it was set to `true` or `false`). This in turn would lead to strange (and inappropriate) errors when using such a string to access entries of a record.
- GAP can store vectors over small finite fields (size at most 256) in a special internal data representation where each entry of the vector uses exactly one byte. Due to an off-by-one bug, the case of a field with exactly 256 elements was not handled properly. As a result, GAP failed to convert a vector to the special data representation, which in some situations could lead to a crash. The off-by-one error was fixed and now vectors over $GF(256)$ work as expected.
- A bug in the code for accessing sublist via the `list{poss}` syntax could lead to GAP crashing. Specifically, if the list was a compressed vector over a finite field, and the sublist syntax was nested, as in `vec{poss1}{poss2}`. This now correctly triggers an error instead of crashing.

New packages added for the redistribution with GAP:

- SpinSym package by L. Maas, which contains Brauer tables of Schur covers of symmetric and alternating groups and provides some related functionalities.

3.3 GAP 4.6.4 (April 2013)

New functionality:

- New command line option `-O` was introduced to disable loading obsolete variables. This option may be used, for example, to check that they are not used in a GAP package or one's own GAP code. For further details see **Reference: options** and **Reference: Replaced and Removed Command Names**.

Fixed bugs which could lead to incorrect results:

- Fixed the bug in `NewmanInfinityCriterion` (**Reference: NewmanInfinityCriterion**) which may cause returning `true` instead of `false`. [Reported by Lev Glebsky]

Fixed bugs which could lead to crashes:

- Fixed the kernel method for `Remove` (**Reference: Remove**) which did not raise an error in case of empty lists, but corrupted the object. The error message in a library method is also improved. [Reported by Roberto Ràdina]

Fixed bugs that could lead to break loops:

- Fixed requirements in a method to multiply a list and an algebraic element. [Reported by Sebastian Gutsche]
- Fixed a bug in `NaturalCharacter` (**Reference: NaturalCharacter (for a group)**) entering a break loop when being called on a homomorphism whose image is not a permutation group. [Reported by Sebastian Gutsche]
- Fixed a bug in `ExponentsConjugateLayer` (**Reference: ExponentsConjugateLayer**) which occured, for example, in some calls of `SubgroupsSolvableGroup` (**Reference: SubgroupsSolvableGroup**) [Reported by Ramon Esteban-Romero]
- Fixed a problem with displaying function fields, e.g. `Field(Indeterminate(Rationals,"x"))`. [Reported by Jan Willem Knopper]
- Fixed two bugs in the code for `NaturalHomomorphismByIdeal` (**Reference: NaturalHomomorphismByIdeal**) for polynomial rings. [Reported by Martin Leuner]
- Added missing method for `String` (**Reference: String**) for `-infinity`.
- Fixed the bug with `ONanScottType` (**Reference: ONanScottType**) not recognising product action properly in some cases.
- The method for `SlotUsagePattern` (**Reference: SlotUsagePattern**) for straight line programs had a bug which triggered an error, if the straight line program contained unnecessary steps.

3.4 GAP 4.6.5 (July 2013)

Improved functionality:

- `TraceMethods` (**Reference: TraceMethods (for operations)**) and `UntraceMethods` (**Reference: UntraceMethods (for operations)**) now better check their arguments and provide a sensible error message if being called without arguments. Also, both variants of calling them are now documented.
- Library methods for `Sortex` (**Reference: Sortex**) are now replaced by faster ones using the kernel `SortParallel` (**Reference: SortParallel**) functionality instead of making expensive zipped lists.

Fixed bugs which could lead to incorrect results:

- `IntHexString` (**Reference: IntHexString**) wrongly produced a large integer when there were too many leading zeros. [Reported by Joe Bohanon]

Fixed bugs that could lead to break loops:

- A bug that may occur in some cases while calling `TransitiveIdentification` (**Reference: TransitiveIdentification**). [Reported by Izumi Miyamoto]
- The new code for semidirect products of permutation groups, introduced in GAP 4.6, had a bug which was causing problems for `Projection` (**Reference: Projection**). [Reported by Graham Ellis]

Chapter 4

Changes between GAP 4.4 and GAP 4.5

This chapter lists most important changes between GAP 4.4.12 and the first public release of GAP 4.5. It also contains information about subsequent update releases for GAP 4.5. It is not meant to serve as a complete account on all improvements; instead, it should be viewed as an introduction to GAP 4.5, accompanying its release announcement.

4.1 Changes in the core GAP system introduced in GAP 4.5

In this section we list most important new features and bugfixes in the core system introduced in GAP 4.5. For the list of changes in the interface between the core system and packages as well as for an overview of new and updated packages, see Section 4.2.

4.1.1 Improved functionality

Performance improvements:

- The GAP kernel now uses GMP (GNU multiple precision arithmetic library, `http://gmplib.org/`) for faster large integer arithmetic.
- Improved performance for records with large number of components.
- Speedup of hash tables implementation at the GAP library level.
- `MemoryUsage` (**Reference: MemoryUsage**) is now much more efficient, in particular for large objects.
- Speedups in the computation of low index subgroups, Tietze transformations, calculating high powers of matrices over finite fields, `Factorial` (**Reference: Factorial**), etc.

New and improved kernel functionality:

- By default, the GAP kernel compiles with the GMP and readline libraries. The GMP library is supplied with GAP and we recommend that you use the version we supply. There are some problems with some other versions. It is also possible to compile the GAP kernel with the system GMP if your system has it. The readline library must be installed on your system in advance to be used with GAP.

- Floating point literals are now supported in the GAP language, so that, floating point numbers can be entered in GAP expressions in a natural way. Support for floats is now properly documented, see (**Reference: Floats**). GAP has an interface using which packages may add new floating point implementations and integrate them with the parser. In particular, we expect that there will soon be a package that implements arbitrary precision floating point arithmetic.
- The Mersenne twister random number generator has been made independent of endianness, so that random seeds can now be transferred between architectures. See (**Reference: GlobalMersenneTwister**) for details.
- Defaults for `-m` and `-o` options have been increased. Changes have been made to the way that GAP obtains memory from the Operating System, to make GAP more compatible with C libraries. A new `-s` option has been introduced to control or turn off the new behaviour, see (**Reference: Command Line Options**).
- The filename and lines from which a function was read can now be recovered using `FilenameFunc` (**Reference: FilenameFunc**), `StartlineFunc` (**Reference: StartlineFunc**) and `EndlineFunc` (**Reference: EndlineFunc**). This allows you, for example, to implement a function such as `PageSource` (**Reference: PageSource**) to show the file containing the source code of a function or a method in a pager, see `Pager` (**Reference: Pager**).
- `CallFuncList` (**Reference: CallFuncList**) was made into an operation so that it can be used to define behaviour of a non-function when called as a function.
- Improvements to the cyclotomic number arithmetic for fields with large conductors.
- Better and more flexible viewing of some large objects.
- Opportunity to interrupt some long kernel computations, e.g. multiplication of compressed matrices, intercepting `Ctrl-C` in designated places in the kernel code by means of a special kernel function for that purpose.
- `ELM_LIST` now allows you to install methods where the second argument is NOT a positive integer.
- Kernel function `DirectoryContents` (**Reference: DirectoryContents**) to get the list of names of files and subdirectories in a directory.
- Kernel functions for Kronecker product of compressed matrices, see `KroneckerProduct` (**Reference: KroneckerProduct**).

New and improved library functionality:

- Extensions of data libraries:
  - Functions and iterators are now available to create and enumerate simple groups by their order up to isomorphism: `SimpleGroup` (**Reference: SimpleGroup**), `SmallSimpleGroup` (**Reference: SmallSimpleGroup**), `SimpleGroupsIterator` (**Reference: SimpleGroupsIterator**) and `AllSmallNonabelianSimpleGroups` (**Reference: AllSmallNonabelianSimpleGroups**).
  - See also packages CTblLib, IRREDSOL and Smallsemi listed in Section 4.2.2.

- Many more methods are now available for the built-in floating point numbers, see (**Reference: Floats**).
- The bound for the proper primality test in `IsPrimeInt` (**Reference: IsPrimeInt**) increased up to 10^{18}.
- Improved code for determining transversal and double coset representatives in large groups.
- Improvements in `Normalizer` (**Reference: Normalizer**) for S_n.
- Smith normal form of a matrix may be computed over arbitrary euclidean rings, see `NormalFormIntMat` (**Reference: NormalFormIntMat**).
- Improved algorithms to determine the subgroup lattice of a group, as well as the function `DotFileLatticeSubgroups` (**Reference: DotFileLatticeSubgroups**) to save the lattice structure in `.dot` file to view it e.g. with GraphViz.
- Special teaching mode which simplifies some output and provides more basic functionality, see (**Reference: Teaching Mode**).
- Functionality specific for use in undergraduate abstract algebra courses, e.g. checksums ((**Reference: Check Digits**)); string/integer list conversion; rings of small orders; the function `SetNameObject` (**Reference: SetNameObject**) to set display names for objects for more informative examples, e.g. constructing groups from "named" objects, such as, for example, `R90` for a 90-degree rotation).
- Functions `DirectoryDesktop` (**Reference: DirectoryDesktop**) and `DirectoryHome` (**Reference: DirectoryHome**) which provide uniform access to default directories under Windows, Mac OS X and Unix.
- Improved methods for hashing when computing orbits.
- Functionality to call external binaries under Windows.
- Symplectic groups over residue class rings, see `SymplecticGroup` (**Reference: SymplecticGroup**).
- Basic version of the simplex algorithm for matrices.
- New functions, operations and attributes: `PrimeDivisors` (**Reference: PrimeDivisors**), `Shuffle` (**Reference: Shuffle**) for lists, `IteratorOfPartitions` (**Reference: IteratorOfPartitions**), `IteratorOfCombinations` (**Reference: IteratorOfCombinations**), `EnumeratorOfCombinations` (**Reference: EnumeratorOfCombinations**) and others.
- The behaviour of `Info` (**Reference: Info**) statements can now be configured per info class, this applies to the way the arguments are printed and to the output stream, see (**Reference: Info Functions**).
- New function `Test` (**Reference: Test**) which is a more flexible and informative substitute of `ReadTest` (**Reference: ReadTest**) operation.
- `ConnectGroupAndCharacterTable` is replaced by more robust function `CharacterTableWithStoredGroup` (**Reference: CharacterTableWithStoredGroup**).

Many problems in GAP have have been fixed, among them the following:

- Polynomial factorisation over rationals could miss factors of degree greater than $deg(f)/2$ if they have very small coefficients, while the cofactor has large coefficients.
- `IntermediateSubgroups` (**Reference: IntermediateSubgroups**) called on a group and a normal subgroup did not properly calculate maximal inclusion relationships.
- `CentreOfCharacter` (**Reference: CentreOfCharacter**) and `ClassPositionsOfCentre` (**Reference: ClassPositionsOfCentre (for a character)**) called for a group character could return a perhaps too large result.
- `Trace` (**Reference: Traces of field elements and matrices**) called for an element of a finite field that was created with `AlgebraicExtension` (**Reference: AlgebraicExtension**) ran into an error.
- `IrreducibleRepresentationsDixon` (**Reference: IrreducibleRepresentationsDixon**) did not accept a list with one character as a second argument.
- Composing a homomorphism from a permutation group to a finitely presented group with another homomorphism could give wrong results.
- For certain arguments, the function `EU` (**Reference: EU**) returned wrong results.
- In the table of marks of cyclic groups, `NormalizersTom` (**Reference: NormalizersTom**) value was wrong.
- The function `PermChars` (**Reference: PermChars**) returned a perhaps wrong result when the second argument was a positive integer (not a record) and the trivial character of the character table given as the first argument was not the first in the list of irreducibles.
- GAP crashed when the intersection of ranges became empty.
- `IsPSL`, and in turn `StructureDescription` (**Reference: StructureDescription**), erroneously recognised non-PSL groups of the right order as PSL.
- The semidirect product method for pcgs computable groups sometimes tried to use finite presentations which were not polycyclic. This usually happened when the groups were not pc groups, and there was a very low risk of getting a wrong result.
- The membership test for a group of finite field elements ran into an error if the zero element of the field was given as the first argument.
- Constant polynomials were not recognised as univariate in any variable.
- The kernel recursion depth counter was not reset properly when running into many break loops.
- GAP did not behave well when printing of a (large) object was interrupted with `Ctrl-C`. Now the object is no longer corrupted and the indentation level is reset.

Potentially incompatible changes:

- The zero polynomial now has degree `-infinity`, see `DegreeOfLaurentPolynomial` (**Reference: DegreeOfLaurentPolynomial**).

- Multiple unary + or - signs are no longer allowed (to avoid confusion with increment/decrement operators from other programming languages).
- Due to changes to improve the performance of records with large number of components, the ordering of record components in `View`'ed records has changed.
- Due to improvements for vectors over finite fields, certain objects have more limitations on changing their base field. For example, one can not create a compressed matrix over $GF(2)$ and then assign an element of $GF(4)$ to one of its entries.

No longer supported:

- Completion files mechanism.
- GAP 3 compatibility mode.

In addition, we no longer recommend using the GAP compiler `gac` to compile GAP code to C, and may withdraw it in future releases. Compiling GAP code only ever gave a substantial speedup for rather specific types of calculation, and much more benefit can usually be achieved quite easily by writing a small number of key functions in C and loading them into the kernel as described in `LoadDynamicModule` (**Reference: LoadDynamicModule**). The `gac` script will remain available as a convenient way of compiling such kernel modules from C.

Also, the following functions and operations were made obsolete: `AffineOperation`, `AffineOperationLayer`, `FactorCosetOperation`, `DisplayRevision`, `ProductPol`, `TeXObj`, `LaTeXObj`.

4.1.2 Changes in distribution formats

The GAP 4.5 source distribution has the form of a single archive containing the core system and the most recent "stable" versions of all currently redistributed packages. There are no optional archives to download: the TomLib package now contains all its tables of marks in one archive; we do not provide separate versions of manuals for Internet Explorer, and the former `tools` archive is now included as an archive in the `etc` directory. To unpack and install the archive, user the script `etc/install-tools.sh`.

We no longer distribute separate bugfix archives when the core GAP system changes, or updated packages archives when a redistributed package is updated. Instead, the single GAP source distribution archive will be labelled by the version of the core GAP system and also by a timestamp. This archive contains the core system and the stable versions of the relevant packages on that date. To upgrade, you simply replace the whole directory containing the GAP installation, and rebuild binaries for the GAP kernel and packages. For new versions of packages, we will also continue to redistribute individual package archives so it will be possible to update a single package without changing the rest of the GAP installation.

Furthermore, by default GAP will now automatically read a user-specific GAP root directory (unless GAP is called with the `-r` option). All user settings can be made in that directory, so there will be no risk of them being lost during an update (see Section 4.1.3 below for more details). Private packages can also be installed in this directory for the same reason.

There are some changes in archive formats used for the distribution: we continue to provide `.tar.gz`, `.tar.bz2` and `-win.zip` archives. We have added `.zip`, and stopped providing `.zoo` archives. We no longer provide GAP binaries for Mac OS 9 (Classic) any more. For installations from source on Mac OS X one may follow the instructions for UNIX.

With the release of GAP 4.5, we also encourage more users to take advantage of the increasingly mature binary distributions which are now available. These include:

- The binary `rsync` distribution for GAP on Linux PCs with i686 or x86_64 compatible processors provided by Frank Lübeck, see http://www.math.rwth-aachen.de/~Frank.Luebeck/gap/rsync.
- BOB, a tool for Linux and Mac OS X to download and build GAP and its packages from source provided by M. Neunhöffer: http://www-groups.mcs.st-and.ac.uk/~neunhoef/Computer/Software/Gap/bob.html.
- The GAP installer for Windows provided by Alexander Konovalov: http://www.gap-system.org/ukrgap/wininst/.

In the near future, we also hope to have a binary distribution for Mac OS X.

Internally, we now have infrastructure to support more robust and frequent releases, and an improved system to fetch and test new versions of the increasingly large number of packages. The Example package documents technical requirements for packages, many of which are checked automatically by our systems. This will allow us to check the compatibility of packages with the system and with other packages more thoroughly before publishing them on the GAP website.

4.1.3 Improvements to the user interface

By default, GAP now uses the readline library for command line editing. It provides such advantages as working with unicode terminals, nicer handling of long input lines, improved TAB-completion and flexible configuration. For further details, see (**Reference: Editing using the readline library**).

We have extended facilities for user interface customisation. By default GAP automatically scans a user specific GAP root directory (unless GAP is called with the `-r` option). The name of this user specific directory depends on the operating system and is contained in `GAPInfo.UserGapRoot`. This directory can be used to tell GAP about personal preferences, to load some additional code, to install additional packages, or to overwrite some GAP files, see (**Reference: GAP Root Directories**). Instead of a single `.gaprc` file we now use more flexible setup based on two files: `gap.ini` which is read early in the startup process, and `gaprc` which is read after the startup process, but before the first input file given on the command line. These files may be located in the user specific GAP root directory `GAPInfo.UserGapRoot` which by default is the first GAP root directory, see (**Reference: The gap.ini and gaprc files**). For compatibility, the `.gaprc` file is still read if the directory `GAPInfo.UserGapRoot` does not exist. See (**Reference: The former .gaprc file**) for the instructions how to migrate your old setup.

Furthermore, there are functions to deal with user preferences, for example, to specify how GAP's online help is shown or whether the coloured prompt should be used. Calls to set user preferences may appear in the user's `gap.ini` file, as explained in (**Reference: Configuring User preferences**).

In the Windows version, we include a new shell which uses the mintty terminal in addition to the two previously used shells (Windows command line and RXVT). The mintty shell is now recommended. It supports Unicode encoding and has flexible configurations options. Also, GAP under Windows now starts in the `%HOMEDRIVE%%HOMEPATH%` directory, which is the user's home directory. Besides this, a larger workspace is now permitted without a need to modify the Windows registry.

Other changes in the user interface include:

- the command line history is now implemented at the GAP level, it can be stored on quitting a GAP session and reread when starting a new session, see (**Reference: The command line history**).
- `SetPrintFormattingStatus("stdout",false);` may be used to switch off the automatic line breaking in terminal output, see `SetPrintFormattingStatus` (**Reference: SetPrintFormattingStatus**).
- GAP supports terminals with up to 4096 columns (extendable at compile time).
- Directories in `-l` command-line option can now be specified starting with `~/`, see (**Reference: Command Line Options**).
- Large integers are now displayed by a short string showing the first and last few digits, and the threshold to trigger this behaviour is user configurable (call `UserPreference("MaxBitsIntView")` to see the default value).
- The GAP banner has been made more compact and informative.
- `SetHelpViewer` (**Reference: SetHelpViewer**) now supports the Google Chrome browser.
- Multiple matches in the GAP online help are displayed via a function from the Browse package, which is loaded in the default configuration. This feature can be replaced by the known pager using the command

```
SetUserPreference( "browse", "SelectHelpMatches", false );
```

4.1.4 Better documentation

The main GAP manuals have been converted to the GAPDoc format provided by the GAPDoc package by Frank Lübeck and Max Neunhöffer (http://www.math.rwth-aachen.de/~Frank.Luebeck/GAPDoc). This documentation format is already used by many packages and is now recommended for all GAP documentation.

Besides improvements to the documentation layout in all formats (text, PDF and HTML), the new GAP manuals incorporate a large number of corrections, clarifications, additions and updated examples.

We now provide two HTML versions of the manual, one of them with MathJax (http://www.mathjax.org) support for better display of mathematical symbols. Also, there are two PDF versions of the manual - a coloured and a monochrome one.

Several separate manuals now became parts of the GAP Reference manual. Thus, now there are three main GAP manual books:

- *GAP Tutorial*
- *GAP Reference manual*
- *GAP - Changes from Earlier Versions* (this manual)

Note that there is no index file combining these three manuals. Instead of that, please use the GAP help system which will search all of these and about 100 package manuals.

4.2 Packages in GAP 4.5

Here we list most important changes affecting packages and present new or essentially changed packages. For the changes in the core GAP system, see Section 4.1.

4.2.1 Interface between the core system and packages

The package loading mechanism has been improved. The most important new feature is that all dependencies are evaluated in advance and then used to determine the order in which package files are read. This allows GAP to handle cyclic dependencies as well as situations where package A requires package B to be loaded completely before any file of package A is read. To avoid distortions of the order in which packages will be loaded, package authors are strongly discouraged from calling `LoadPackage` (**Reference: LoadPackage**) and `TestPackageAvailability` (**Reference: TestPackageAvailability**) in a package code in order to determine whether some other package will be loaded before or together with the current package - instead, one should use `IsPackageMarkedForLoading` (**Reference: IsPackageMarkedForLoading**). In addition, there is now a better error management if package loading fails for packages that use the new functionality to log package loading messages (see `DisplayPackageLoadingLog` (**Reference: DisplayPackageLoadingLog**) and the rest of the Chapter (**Reference: Using GAP Packages**) which documents how to *use* GAP packages), and package authors are very much encouraged to use these logging facilities.

In GAP 4.4 certain packages were marked as *autoloaded* and would be loaded, if present, when GAP started up. In GAP 4.5, this notion is divided into three. Certain packages are recorded as *needed* by the GAP system and others as *suggested*, in the same way that packages may *need* or *suggest* other packages. If a needed package is not loadable, GAP will not start. Currently only GAPDoc is needed. If a suggested package is loadable, it will be loaded. Typically these are packages which install better methods for Operations and Objects already present in GAP. Finally, the user preferences mechanism can be used to specify additional packages that should be loaded if possible. By default this includes most packages that were autoloaded in GAP 4.4.12, see `ShowUserPreferences` (**Reference: ShowUserPreferences**).

GAP packages may now use local *namespaces* to avoid name clashes for global variables introduced in other packages or in the GAP library, see (**Reference: Namespaces for GAP packages**).

All guidance on how to *develop* a GAP package has been consolidated in the Example package which also contains a checklist for upgrading a GAP package to GAP 4.5, see (**Example: Guidelines for Writing a GAP Package**).

4.2.2 New and updated packages since GAP 4.4.12

At the time of the release of GAP 4.4.12 there were 75 packages redistributed with GAP (including the TomLib which was distributed in the core GAP archive). The first public release of GAP 4.5 contains precisely 99 packages.

The new packages that have been added to the redistribution since the release of GAP 4.4.12 are:

- Citrus package by J.D. Mitchell for computations with transformation semigroups and monoids (this package is a replacement of the Monoid package).
- cvec package by M. Neunhöffer, providing an implementation of compact vectors over finite fields.

- fwtree package by B. Eick and T. Rossmann for computing trees related to some pro-p-groups of finite width.
- GBNP package by A.M. Cohen and J.W. Knopper, providing algorithms for computing Grobner bases of noncommutative polynomials over fields with respect to the "total degree first then lexicographical" ordering.
- genss package by M. Neunhöffer and F. Noeske, implementing the randomised Schreier-Sims algorithm to compute a stabiliser chain and a base and a strong generating set for arbitrary finite groups.
- HAPprime package by P. Smith, extending the HAP package with an implementation of memory-efficient algorithms for the calculation of resolutions of small prime-power groups.
- hecke package by D. Traytel, providing functions for calculating decomposition matrices of Hecke algebras of the symmetric groups and q-Schur algebras (this package is a port of the GAP 3 package Specht 2.4 to GAP 4).
- Homalg project by M. Barakat, S. Gutsche, M. Lange-Hegermann et al., containing the following packages for the homological algebra: homalg, ExamplesForHomalg, Gauss, GaussForHomalg, GradedModules, GradedRingForHomalg, HomalgToCAS, IO_ForHomalg, LocalizeRingForHomalg, MatricesForHomalg, Modules, RingsForHomalg and SCO (see http://homalg.math.rwth-aachen.de/).
- MapClass package by A. James, K. Magaard and S. Shpectorov to calculate the mapping class group orbits for a given finite group.
- recogbase package by M. Neunhöffer and A. Seress, providing a framework to implement group recognition methods in a generic way (suitable, in particular, for permutation groups, matrix groups, projective groups and black box groups).
- recog package by M. Neunhöffer, A. Seress, N. Ankaralioglu, P. Brooksbank, F. Celler, S. Howe, M. Law, S. Linton, G. Malle, A. Niemeyer, E. O'Brien and C.M. Roney-Dougal, extending the recogbase package and provides a collection of methods for the constructive recognition of groups (mostly intended for permutation groups, matrix groups and projective groups).
- SCSCP package by A. Konovalov and S. Linton, implementing the Symbolic Computation Software Composability Protocol (SCSCP, see http://www.symbolic-computation.org/scscp) for GAP, which provides interfaces to link a GAP instance with another copy of GAP or other SCSCP-compliant system running locally or remotely.
- simpcomp package by F. Effenberger and J. Spreer for working with simplicial complexes.
- Smallsemi package by A. Distler and J.D. Mitchell, containing the data library of all semigroups with at most 8 elements as well as various information about them.
- SymbCompCC package by D. Feichtenschlager for computations with parametrised presentations for finite p-groups of fixed coclass.

Furthermore, some packages have been upgraded substantially since the GAP 4.4.12 release:

- Alnuth package by B. Assmann, A. Distler and B. Eick uses an interface to PARI/GP system instead of the interface to KANT (thanks to B. Allombert for the GP code for the new interface and help with the transition) and now also works under Windows.
- CTblLib package (the GAP Character Table Library) by T. Breuer has been extended by many new character tables, a few bugs have been fixed, and new features have been added, for example concerning the relation to GAP's group libraries, better search facilities, and interactive overviews. For details, see the package manual.
- DESIGN package by L.H. Soicher:
  - The functions `PointBlockIncidenceMatrix`, `ConcurrenceMatrix`, and `InformationMatrix` compute matrices associated with block designs.
  - The function `BlockDesignEfficiency` computes certain statistical efficiency measures of a $1-(v,k,r)$ design, using exact algebraic computation.
- Example package by W. Nickel, G. Gamble and A. Konovalov has a more detailed and up-to-date guidance on developing a GAP package, see (**Example: Guidelines for Writing a GAP Package**).
- FR package by L. Bartholdi now uses floating-point numbers to compute approximations of rational maps given by their group-theoretical description.
- The GAPDoc package by F. Lübeck and M. Neunhöffer provides various improvements, for example:
  - The layout of the text version of the manuals can be configured quite freely, several standard "themes" are provided. The display is now adjusted to the current screen width.
  - Some details of the layout of the HTML version of the manuals can now be configured by the user. All manuals are available with and without MathJax support for display of mathematical formulae.
  - The text and HTML versions of manuals make more use of unicode characters (but the text version is also still reasonably good on terminals with latin1 or ASCII encoding).
  - The PDF version of the manuals uses better fonts.
  - Of course, there are various improvements for authors of manuals as well, for example new functions `ExtractExamples` (**GAPDoc: ExtractExamples**) and `RunExamples` (**GAPDoc: RunExamples**) for automatic testing and correcting of manual examples.
- Gpd package by E.J. Moore and C.D. Wensley has been substantially rewritten. The main extensions provide functions for:
  - Subgroupoids of a direct product with complete graph groupoid, specified by a root group and choice of rays.
  - Automorphisms of finite groupoids - by object permutations; by root group automorphisms; and by ray images.
  - The automorphism group of a finite groupoid together with an isomorphism to a quotient of permutation groups.

  - Homogeneous groupoids (unions of isomorphic groupoids) and their morphisms, in particular homogeneous discrete groupoids: the latter are used in constructing crossed modules of groupoids in the XMod package.

- GRAPE package by L.H. Soicher:
  - With much help from A. Hulpke, the interface between GRAPE and `dreadnaut` is now done entirely in GAP code.
  - A 32-bit `nauty/dreadnaut` binary for Windows (XP and later) is included with GRAPE, so now GRAPE provides full functionality under Windows, with no installation necessary.
  - Graphs with ordered partitions of their vertices into "colour-classes" are now handled by the graph automorphism group and isomorphism testing functions. An automorphism of a graph with colour-classes is an automorphism of the graph which additionally preserves the list of colour-classes (classwise), and an isomorphism from one graph with colour-classes to a second is a graph isomorphism from the first graph to the second which additionally maps the first list of colour-classes to the second (classwise).
  - The GAP code and old standalone programs for the undocumented functions `Enum` and `EnumColadj` have been removed as their functionality can now largely be handled by current documented GAP and GRAPE functions.

- IO package by M. Neunhöffer:
  - New build system to allow for more flexibility regarding the use of compiler options and adjusting to GAP 4.5.
  - New functions to access time like `IO_gettimeofday`, `IO_gmtime` and `IO_localtime`.
  - Some parallel skeletons built on `fork` like: `ParListByFork`, `ParMapReduceByFork`, `ParTakeFirstResultByFork` and `ParWorkerFarmByFork`.
  - `IOHub` objects for automatic I/O multiplexing.
  - New functions `IO_gethostbyname` and `IO_getsockname`.

- IRREDSOL package by B. Höfling now covers all irreducible soluble subgroups of $GL(n,q)$ for $q^n < 1000000$ and primitive soluble permutation groups of degree < 1000000 (previously, the bound was 65536). It also has faster group recognition and adds a few omissions for $GL(3,8)$ and $GL(6,5)$.

- ParGAP package by G. Cooperman is now compiled using a system-wide MPI implementation by default to facilitate running it on proper clusters. There is also an option to build it with the MPINU library which is still supplied with the package (thanks to P. Smith for upgrading ParGAP build process).

- OpenMath package by M. Costantini, A. Konovalov, M. Nicosia and A. Solomon now supports much more OpenMath symbols to facilitate communication by the remote procedure call protocol implemented in the SCSCP package. Also, a third-party external library to support binary OpenMath encoding has been replaced by a proper implementation made entirely in GAP.

- Orb package by J. Müller, M. Neunhöffer and F. Noeske:

  There have been numerous improvements to this package:

  - A new fast implementation of AVL trees (balanced binary trees) in C.
  - New interface to hash table functionality and implementation in C for speedup.
  - Some new hash functions for various object types like transformations.
  - New function `ORB_EstimateOrbitSize` using the birthday paradox.
  - Improved functionality for product replacer objects.
  - New "tree hash tables".
  - New functionality to compute weak and strong orbits for semigroups and monoids.
  - `OrbitGraph` for Orb orbits.
  - Fast C kernel methods for the following functions: `PermLeftQuoTransformationNC`, `MappingPermSetSet`, `MappingPermListList`, `ImageSetOfTransformation`, and `KernelOfTransformation`.
  - New build system to allow for more flexibility regarding the use of compiler options and to adjust to GAP 4.5.

- RCWA package by S. Kohl among the new features and other improvements has the following:
  - A database of all 52394 groups generated by 3 class transpositions of $\mathbb{Z}$ which interchange residue classes with modulus less than or equal to 6. This database contains the orders and the moduli of all of these groups. Also it provides information on what is known about which of these groups are equal and how their finite and infinite orbits on $\mathbb{Z}$ look like.
  - More routines for investigating the action of an rcwa group on $\mathbb{Z}$. Examples are a routine which attempts to find out whether a given rcwa group acts transitively on the set of nonnegative integers in its support and a routine which looks for finite orbits on the set of all residue classes of $\mathbb{Z}$.
  - Ability to deal with rcwa permutations of $\mathbb{Z}^2$.
  - Important methods have been made more efficient in terms of runtime and memory consumption.
  - The output has been improved. For example, rcwa permutations are now `Display`'ed in ASCII text resembling LaTeX output.

- The XGAP package by F. Celler and M. Neunhöffer can now be used on 64-bit architectures (thanks to N. Eldredge and M. Horn for sending patches). Furthermore, there is now an export to XFig option (thanks to Russ Woodroofe for this patch). The help system in XGAP has been adjusted to GAP 4.5.

- Additionally, some packages with kernel modules or external binaries are now available in Windows. The `-win.zip` archive and the GAP installer for Windows include working versions of the following packages: Browse, cvec, EDIM, GRAPE, IO and orb, which were previously unavailable for Windows users.

Finally, the following packages are withdrawn:

- IF package by M. Costantini is unmaintained and no longer usable. More advanced functionality for interfaces to other computer algebra systems is now available in the SCSCP package by A. Konovalov and S. Linton.

- Monoid package by J. Mitchell is superseded by the Citrus package by the same author.

- NQL package by R. Hartung has been withdrawn by the author.

4.3 GAP 4.5.5 (July 2012)

Fixed bugs which could lead to crashes:

- For small primes (compact fields) `ZmodnZObj(r,p)` now returns the corresponding FFE to avoid crashes when compacting matrices. [Reported by Ignat Soroko]

Other fixed bugs:

- Fixed a bug in `CommutatorSubgroup` (**Reference: CommutatorSubgroup**) for fp groups causing infinite recursion, which could, for example, be triggered by computing automorphism groups.
- Previously, the list of factors of a polynomial was mutable, and hence could be accidentally corrupted by callers. Now the list of irreducible factors is stored immutable. To deal with implicit reliance on old code, always a shallow copy is returned. [reported by Jakob Kroeker]
- Computing high powers of matrices ran into an error for matrices in the format of the cvec package. Now the library function also works with these matrices. [reported by Klaus Lux]
- The pseudo tty code which is responsible for spawning subprocesses has been partially rewritten to allow more than 128 subprocesses on certain systems. This mechanism is for example used by ANUPQ and nq packages to compute group quotients via an external program. Previously, on Mac OS X this could be done precisely 128 times, and then an error would occur. That is, one could e.g. compute 128 nilpotent quotients, and then had to restart GAP to compute more. This also affected other systems, such as OpenBSD, where it now also works correctly.
- On Mac OS X, using GAP compiled against GNU readline 6.2, pasting text into the terminal session would result in this text appearing very slowly, with a 0.1 sec delay between each "keystroke". This is not the case with versions 6.1 and older, and has been reported to the GNU readline team. In the meantime, we work around this issue in most situations by setting `rl_event_hook` only if `OnCharReadHookActive` is set.
- `ShowUserPreferences` (**Reference: ShowUserPreferences**) ran into a break loop in case of several undeclared user preferences. [Reported by James Mitchell]
- GAP did not start correctly if the user preference `"InfoPackageLoadingLevel"` was set to a number $>= 3$. The reason is that `PrintFormattedString` was called before it was installed. The current fix is a temporary solution.
- The `"hints"` member of `TypOutputFile` used to contain 3*100 entries, yet `addLineBreakHint` would write entries with index up to and including 3*99+3=300, leading to a buffer overflow. This would end up overwriting the `"stream"` member with -1. Fixed by incrementing the size of `"hints"` to 301. [Reported by Jakob Kroeker]
- The function `IsDocumentedWord` tested the given word against strings obtained by splitting help matches at non-letter characters. This way, variable names containing underscores or digits were erroneously not regarded as documented, and certain substrings of these names were erroneously regarded as documented.

- On Windows, an error occurred if one tried to use the default Windows browser as a help viewer (see `SetHelpViewer` (**Reference: SetHelpViewer**)). Now the browser opens the top of the correspoding manual chapter. The current fix is a temporary solution since the problem remains with the positioning at the required manual section.

Improved functionality:

- `WriteGapIniFile` (**Reference: WriteGapIniFile**) on Windows now produces the `gap.ini` file with Windows style line breaks. Also, an info message is now printed if an existing `gap.ini` file was moved to a backup file `gap.ini.bak`.
- The CTblLib and TomLib packages are removed from the list of suggested packages of the core part of GAP. Instead they are added to the default list of the user preference `"PackagesToLoad"`. This way it is possible to configure GAP to not load these packages via changing the default value of `"PackagesToLoad"`.
- The conjugacy test in S_n for intransitive subgroups was improved. This deals with inefficiency issue in the case reported by Stefan Kohl.
- Added `InstallAndCallPostRestore` to `lib/system.g` and call it in `lib/init.g` instead of `CallAndInstallPostRestore` for the function that reads the files listed in GAP command line. This fixes the problem reported by Yevgen Muntyan when `SaveWorkspace` (**Reference: SaveWorkspace**) was used in a file listed in GAP command line (before, according to the documentation, `SaveWorkspace` (**Reference: SaveWorkspace**) was only allowed at the main GAP prompt).
- There is now a new user preference `PackagesToIgnore`, see `SetUserPreference` (**Reference: SetUserPreference**). It contains a list of names of packages that shall be regarded as not available at all in the current session, both for autoloading and for later calls of `LoadPackage` (**Reference: LoadPackage**). This preference is useful for testing purposes if one wants to run some code without loading certain packages.

4.4 GAP 4.5.6 (September 2012)

Improved functionality:

- The argument of `SaveWorkspace` (**Reference: SaveWorkspace**) can now start with `~/` which is expanded to the users home directory.
- Added the method for `Iterator` (**Reference: Iterator**) for `PositiveIntegers` (**Reference: PositiveIntegers**). [Suggested by Attila Egri-Nagy].
- Changed kernel tables such that list access functionality for `T_SINGULAR` objects can be installed by methods at the GAP level.
- In case of saved history, "UP" arrow after starting GAP yields last stored line. The user preference `HistoryMaxLines` is now used when storing and saving history (see `SetUserPreference` (**Reference: SetUserPreference**)).

Fixed bugs which could lead to crashes:

- A crash occuring during garbage collection following a call to `AClosVec` for a `GF(2)` code. [Reported by Volker Braun]
- A crash when parsing certain syntactically invalid code. [Reported by multiple users]
- Fixed and improved command line editing without readline support. Fixed a segfault which could be triggered by a combination of "UP" and "DOWN" arrows. [Reported by James Mitchell]
- Fixed a bug in the kernel code for floats that caused a crash on SPARC Solaris in 32-bit mode. [Reported by Volker Braun]

Other fixed bugs:

- Very large (more than 1024 digit) integers were not being coded correctly in function bodies unless the integer limb size was 16 bits. [Reported by Stefan Kohl]
- An old variable was used in assertion, causing errors in a debugging compilation. [Reported by Volker Braun]
- The environment variable `PAGER` is now correctly interpreted when it contains the full path to the pager program. Furthermore, if the external pager `less` is found from the environment it is made sure that the option `-r` is used (same for `more -f`). [Reported by Benjamin Lorenz]
- Fixed a bug in `PermliftSeries`. [Reported by Aiichi Yamasaki]
- Fixed discarder function in lattice computation to distinguish general and zuppo discarder. [Reported by Leonard Soicher]
- The `GL` (**Reference: GL (for dimension and a ring)**) and `SL` (**Reference: SL (for dimension and a ring)**) constructors did not correctly handle `GL(filter,dim,ring)`.
- The names of two primitive groups of degree 64 were incorrect.
- The `\in` (**Reference: \in operation for testing membership**) method for groups handled by a nice monomorphism sometimes could produce an error in situations where it should return false. This only happened when using `SeedFaithfulAction` to influence how `NiceMonomorphism` (**Reference: NiceMonomorphism**) builds the nice monomorphims for a matrix groups.
- Wrong `PrintObj` (**Reference: PrintObj**) method was removed to fix delegations accordingly to (**Reference: View and Print**).
- Fixed a method for `Coefficients` (**Reference: Coefficients**) which, after Gaussian elimination, did not check that the coefficients actually lie in the left-acting-domain of the vector space. This could lead to a wrong answer in a vector space membership test. [Reported by Kevin Watkins]

Improved documentation:

- Removed outdated statements from the documentation of `StructureDescription` (**Reference: StructureDescription**) which now non-ambiguosly states that `StructureDescription` is not an isomorphism invariant: non-isomorphic groups can have the same string value, and two isomorphic groups in different representations can produce different strings.

- GAP now allows overloading of a loaded help book by another one. In this case, only a warning is printed and no error is raised. This makes sense if a book of a not loaded package is loaded in a workspace and then GAP is started with a root path that contains a newer version. [Reported by Sebastian Gutsche]
- Provided a better description of user preferences mechanism ((**Reference: Configuring User preferences**)) and a hint to familiarise with them using `WriteGapIniFile` (**Reference: WriteGapIniFile**) function to create a file which contains descriptions of all known user preferences and also sets those user preferences which currently do not have their default value. One can then edit that file to customize (further) the user preferences for future GAP sessions.

New packages added for the redistribution with GAP:

- AutoDoc package by S. Gutsche, providing tools for automated generation of GAPDoc manuals.
- Convex package by S. Gutsche, which provides structures and algorithms for convex geometry.
- PolymakeInterface package by T. Baechler and S. Gutsche, providing a link to the callable library of the polymake system (`http://www.polymake.org`).
- ToolsForHomalg package by M. Barakat, S. Gutsche and M. Lange-Hegermann, which provides some auxiliary functionality for the homalg project (`http://homalg.math.rwth-aachen.de/`).

4.5 GAP 4.5.7 (December 2012)

Fixed bugs which could lead to crashes:

- Closing with `LogInputTo` (or `LogOutputTo`) a logfile opened with `LogTo` (**Reference: LogTo**) left the data structures corrupted, resulting in a crash.
- On 32-bit systems we can have long integers `n` such that `Log2Int(n)` is not an immediate integer. In such cases `Log2Int` gave wrong or corrupted results which in turn could crash GAP, e.g., in `ViewObj(n)`.
- Some patterns of use of `UpEnv` (**Reference: UpEnv**) and `DownEnv` (**Reference: DownEnv**) were leading to a segfault.

Other fixed bugs:

- Viewing of long negative integers was broken, because it went into a break loop.
- Division by zero in `ZmodnZ` (**Reference: ZmodnZ**) (n not prime) produced invalid objects. [Reported by Mark Dickinson]
- Fixed a bug in determining multiplicative inverse for a zero polynomial.
- Fixed a bug causing infinite recursion in `NaturalHomomorphismByNormalSubgroup` (**Reference: NaturalHomomorphismByNormalSubgroup**).

- A workaround was added to deal with a package method creating pcgs for permutation groups for which the entry `permpcgsNormalSteps` is missing.
- For a semigroup of associative words that is not the full semigroup of all associative words, the methods for `Size` (**Reference: Size**) and `IsTrivial` (**Reference: IsTrivial**) called one another causing infinite recursion.
- The 64-bit version of the `gac` script produced wrong (>= 2^31) CRC values because of an integer conversion problem.
- It was not possible to compile GAP on some systems where `HAVE_SELECT` detects as false.
- Numbers in memory options on the command line exceeding 2^32 could not be parsed correctly, even on 64-bit systems. [Reported by Volker Braun]

New packages added for the redistribution with GAP:

- Float package by L. Bartholdi, which extends GAP floating-point capabilities by providing new floating-point handlers for high-precision real, interval and complex arithmetic using MPFR, MPFI, MPC or CXSC external libraries. It also contains a very high-performance implementation of the LLL (Lenstra-Lenstra-Lovász) lattice reduction algorithm via the external library FPLLL.
- ToricVarieties package by S. Gutsche, which provides data structures to handle toric varieties by their commutative algebra structure and by their combinatorics.

Chapter 5

Overview of updates for GAP 4.4

This chapter lists changes in the main system (excluding packages) that have been corrected or added in bugfixes and updates for GAP 4.4.

5.1 GAP 4.4 Bugfix 2 (April 2004)

Fixed bugs which could lead to crashes:

- A crash when incorrect types of arguments are passed to `FileString`.

Other fixed bugs:

- A bug in `DerivedSubgroupTom` and `DerivedSubgroupsTom`.
- An error in the inversion of certain `ZmodnZObj` elements.
- A wrong display string of the numerator in rational functions returned by `MolienSeriesWithGivenDenominator` (in the case that the constant term of this numerator is zero).

5.2 GAP 4.4 Bugfix 3 (May 2004)

Fixed bugs which could produce wrong results:

- Incorrect setting of system variables (e.g., home directory and command line options) after loading a workspace.
- Wrong handling of integer literals within functions or loops on 64-bit architectures (only integers in the range from 2^{28} to 2^{60}).

Fixed bugs which could lead to crashes:

- A problem in the installation of the multiplication routine for matrices that claimed to be applicable for more general list multiplications.
- A problem when computing weight distributions of codes with weights > 2^{28}.

Other fixed bugs:

- Problems with the online help with some manual sections.
- Problems of the online help on Windows systems.
- A problem in `GQuotients` when mapping from a finitely presented group which has a free direct factor.
- A bug in the function `DisplayRevision`.
- The trivial finitely presented group on no generators was not recognized as finitely presented.
- A problem with `Process`.
- A problem when intersecting subgroups of finitely presented groups that are represented in "quotient representation" with the quotient not apermutation group.
- A bug in the generic `Intersection2` method for vector spaces, in the case that both spaces are trivial.
- Enable ReeGroup(q) for $q = 3$.

5.3 GAP 4.4 Bugfix 4 (December 2004)

Fixed bugs which could produce wrong results:

- An error in the `Order` method for matrices over cyclotomic fields which caused this method to return `infinity` for matrices of finite order in certain cases.
- Representations computed by `IrreducibleRepresentations` in characteristic 0 erraneously claimed to be faithful.
- A primitive representation of degree 574 for PSL(2,41) has been missing in the classification on which the GAP library was built.
- A bug in `Append` for compressed vectors over GF(2): if the length of the result is 1 mod 32 (or 64) the last entry was forgotten to copy.
- A problem with the Ree group Ree(3) of size 1512 claiming to be simple.
- An error in the membership test for groups GU(n,q) and SU(n,q) for non-prime q.
- An error in the kernel code for ranges which caused e.g. `-1 in [1..2]` to return `true`.
- An error recording boolean lists in saved workspaces.
- A problem in the selection function for primitive and transitive groups if no degree is given.
- `ReducedConfluentRewritingSystem` returning a cached result that might not conform to the ordering specified.

Other fixed bugs:

- A problem with the function `SuggestUpdates` to check for the most recent version of packages available.

- A problem that caused `MatrixOfAction` to produce an error when the algebra module was constructed as a direct sum.
- Problems with computing n-th power maps of character tables, where n is negative and the table does not yet store its irreducible characters.
- Element conjugacy in large-base permutation groups sometimes was unnecessarily inefficient.
- A missing method for getting the letter representation of an associate word in straight line program representation.
- A problem with the construction of vector space bases where the given list of basis vectors is itself an object that was returned by `Basis`.
- A problem of `AbelianInvariantsMultiplier` insisting that a result of `IsomorphismFpGroup` is known to be surjective.
- An error in the routine for `Resultant` if one of the polynomials has degree zero.

5.4 GAP 4.4 Update 5 (May 2005)

Fixed bugs which could produce wrong results:

- `GroupWithGenerators` (**Reference: GroupWithGenerators**) returned a meaningless group object instead of signaling an error when it was called with an empty list of generators.
- When computing preimages under an embedding into a direct product of permutation groups, if the element was not in the image of the embedding then a permutation had been returned instead of `fail`.
- Two problems with `PowerMod` (**Reference: PowerMod**) for polynomials. [Reported by Jack Schmidt]
- Some methods for computing the sum of ideals returned the first summand instead of the sum. [Reported by Alexander Konovalov]
- Wrong result in `Intersection` (**Reference: Intersection**) for pc groups.
- The function `CompareVersionNumbers` (**Reference: CompareVersionNumbers**) erroneously ignored leading non-digit characters.

  A new feature in the corrected version is an optional third argument `"equal"`, which causes the function to return `true` only if the first two arguments describe equal version numbers; documentation is available in the ext-manual. This new feature is used in `LoadPackage` (**Reference: LoadPackage**), now one can require a specific version of a package.

  The library code still contained parts of the handling of completion files for packages, which does not work and therefore had already been removed from the documentation. This code has now been removed.

  Now a new component `PreloadFile` is supported in `The PackageInfo.g File` (**Reference: The PackageInfo.g File**) files; if it is bound then the file in question is read immediately before the package or its documentation is loaded.

- The result of `String` (**Reference: String**) for strings not in `IsStringRep` (**Reference: IsStringRep**) that occur as list entries or record components was erroneously missing the double quotes around the strings.
- A bug which caused `InducedPcgs` (**Reference: InducedPcgs**) to return a pcgs which is not induced wrt. the parent pcgs of `pcgs`. This may cause unpredictable behaviour, e. g. when `SiftedPcElement` is used subsequently. [Reported by Alexander Konovalov]
- Fixed a bug in `SmallGroupsInformation(512)`.
- `PowerModCoeffs` (**Reference: PowerModCoeffs**) with exponent 1 for compressed vectors did not reduce (a copy of) the input vector before returning it. [Reported by Frank Lübeck]
- Sorting a mutable non-plain list (e.g., a compressed matrix over fields of order < 257) could potentially destroy that object. [Reported by Alexander Hulpke]
- Under rare circumstances computing the closure of a permutation group by a normalizing element could produce a corrupted stabilizer chain. (The underlying algorithm uses random elements, probability of failure was below 1 percent). [Reported by Thomas Breuer]

Fixed bugs which could lead to crashes:

- Some code and comments in the GAP kernel assumed that there is no garbage collection during the core printing function `Pr`, which is not correct. This could cause GAP in rare cases to crash during printing permutations, cyclotomics or strings with zero bytes. [Reported by Warwick Harvey]

Other fixed bugs:

- A rare problem with the choice of prime in the Dixon-Schneider algorithm for computing the character table of a group. [Reported by Jack Schmidt]
- `DirectProduct` (**Reference: DirectProduct**) for trivial permutation groups returned a strange object.
- A problem with `PolynomialReduction` (**Reference: PolynomialReduction**) running into an infinite loop.
- Adding linear mappings with different image domains was not possible. [Reported by Pasha Zusmanovich]
- Multiplying group ring elements with rationals was not possible. [Reported by Laurent Bartholdi]
- `Random` (**Reference: Random Sources**) now works for finite fields of size larger than 2^{28}. [Reported by Jack Schmidt]
- Univariate polynomial creators did modify the coefficient list passed. [Reported by Jürgen Müller]
- Fixed `IntHexString` (**Reference: IntHexString**) to accept arguments not in `IsStringRep`; the argument is now first converted if necessary. [Reported by Kenn Heinrich]

- The library code for stabilizer chains contained quite some explicit references to the identity `()`. This is unfortunate if one works with permutation groups, the elements of which are not plain permutations but objects which carry additional information like a memory, how they were obtained from the group generators. For such cases it is much cleaner to use the `One(...)` operation instead of `()`, such that the library code can be used for a richer class of group objects. This fix contains only rather trivial changes `()` to `One(...)` which were carefully checked by me. The tests for permutation groups all run without a problem. However, it is relatively difficult to provide test code for this particular change, since the "improvement" only shows up when one generates new group objects. This is for example done in the package recog which is in preparation. [Reported by Akos Seress and Max Neunhöffer]

- Using `{}` to select elements of a known inhomogenous dense list produced a list that might falsely claim to be known inhomogenous, which could lead to a segfault if the list typing code tried to mark it homogenous, since the code intended to catch such errors also had a bug. [Reported by Steve Linton]

- The record for the generic iterator construction of subspaces domains of non-row spaces was not complete.

- When a workspace has been created without packages(`-A` option) and is loaded into a GAP session without packages (same option) then an error message is printed.

- So far the functions `IsPrimeInt` (**Reference: IsPrimeInt**) and `IsProbablyPrimeInt` (**Reference: IsProbablyPrimeInt**) are essentially the same except that `IsPrimeInt` issues an additional warning when (non-proven) probable primes are considered as primes.

  These warnings now print the probable primes in question as well; if a probable prime is used several times then the warning is also printed several times; there is no longer a warning for some known large primes; the warnings can be switched off. See `IsPrimeInt` (**Reference: IsPrimeInt**) for more details.

  If we get a reasonable primality test in GAP we will change the definition of `IsPrimeInt` to do a proper test.

- Corrected some names of primitive groups in degree 26. [Reported by Robert F. Bailey]

New or improved functionality:

- Several changes for `ConwayPolynomial` (**Reference: ConwayPolynomial**):
  - many new pre-computed polynomials
  - put data in several separate files (only read when needed)
  - added info on origins of pre-computed polynomials
  - improved performance of `ConwayPolynomial` (**Reference: ConwayPolynomial**) and `IsPrimitivePolynomial` (**Reference: IsPrimitivePolynomial**) for p < 256
  - improved documentation of `ConwayPolynomial`
  - added and documented new functions `IsCheapConwayPolynomial` (**Reference: IsCheapConwayPolynomial**) and `RandomPrimitivePolynomial` (**Reference: RandomPrimitivePolynomial**)

- Added method for `NormalBase` (**Reference: NormalBase**) for extensions of finite fields.
- Added more help viewers for the HTML version of the documentation (firefox, mozilla, konqueror, w3m, safari).
- New function `ColorPrompt` (**Reference: ColorPrompt**). (Users of former versions of a `colorprompt.g` file: Now you just need a `ColorPrompt(true);` in your `.gaprc` file.)
- Specialised kernel functions to support GUAVA 2.0. GAP will only load GUAVA in version at least 2.002 after this update.
- Now there is a kernel function `CYC_LIST` for converting a list of rationals into a cyclotomic, without arithmetics overhead.
- New functions `ContinuedFractionExpansionOfRoot` (**Reference: ContinuedFractionExpansionOfRoot**) and `ContinuedFractionApproximationOfRoot` (**Reference: ContinuedFractionApproximationOfRoot**) for computing continued fraction expansions and continued fraction approximations of real roots of polynomials with integer coefficients.
- A method for computing structure descriptions for finite groups, available via `StructureDescription` (**Reference: StructureDescription**).
- This change contains the new, extended version of the SmallGroups package. For example, the groups of orders p^4, p^5, p^6 for arbitrary primes p, the groups of square-free order and the groups of cube-free order at most 50000 are included now. For more detailed information see the announcement of the extended package.
- The function `ShowPackageVariables` gives an overview of the global variables in a package. It is thought as a utility for package authors and referees. (It uses the new function `IsDocumentedVariable`.)
- The mechanisms for testing GAP has been improved:
  - The information whether a test file belongs to the list in `tst/testall.g` is now stored in the test file itself.
  - Some targets for testing have been added to the `Makefile` in the GAP root directory, the output of the tests goes to the new directory `dev/log`.
  - Utility functions for testing are in the new file `tst/testutil.g`. Now the loops over (some or all) files `tst/*.tst` can be performed with a function call, and the file `tst/testall.g` can be created automatically; the file `tst/testfull.g` is now obsolete. The remormalization of the scaling factors can now be done using a GAP function, so the file `tst/renorm.g` is obsolete.
  - Now the functions `START_TEST` and `STOP_TEST` use components in `GAPInfo` instead of own globals, and the random number generator is always reset in `START_TEST`.
  - `GAPInfo.SystemInformation` now takes two arguments, now one can use it easier in the tests.
- `MultiplicationTable` (**Reference: MultiplicationTable**) is now an attribute, and the construction of a magma, monoid, etc. from multiplication tables has been unified.

5.5 GAP 4.4 Update 6 (September 2005)

Attribution of bugfixes and improved functionalities to those who reported or provided these, respectively, is still fairly incomplete and inconsistent with this update. We apologise for this fact and will discuss until the next update how to improve this feature.

Fixed bugs which could produce wrong results:

- The perfect group library does not contain any information on the trivial group, so the trivial group must be handled specially. `PerfectGroup` (**Reference: PerfectGroup**) and `NrPerfectLibraryGroups` were changed to indicate that the trivial group is not part of the library.
- The descriptions of `PerfectGroup(734832,3)` and `PerfectGroup(864000,3)` were corrected in the `Finite Perfect Groups` (**Reference: Finite Perfect Groups**) library of perfect groups.
- The functions `EpimorphismSchurCover` (**Reference: EpimorphismSchurCover**) and `AbelianInvariantsMultiplier` (**Reference: AbelianInvariantsMultiplier**) may have produced wrong results without warning [Reported by Colin Ingalls]. These problems are fixed. However, the methods currently used can be expected to be slower than the ones used before; we hope to fix this in the next version of GAP.
- `DerivedSubgroup` (**Reference: DerivedSubgroup**) and `CommutatorSubgroup` (**Reference: CommutatorSubgroup**) for permutation groups sometimes returned groups with an incorrect stabilizer chain due to a missing verification step after a random Schreier Sims.
- `NaturalHomomorphismByNormalSubgroup` (**Reference: NaturalHomomorphismByNormalSubgroup**) for FpGroups did unnecessary rewrites.
- The alternating group A_3 incorrectly claimed to be not simple.
- `ExponentSyllable` (**Reference: ExponentSyllable**) for straight line program elements gave a wrong result.
- `PrimePGroup` (**Reference: PrimePGroup**) is defined to return `fail` for trivial groups, but if the group was constructed as a factor or subgroup of a known p-group, the value of p was retained.
- The functions `TestPackageAvailability` (**Reference: TestPackageAvailability**) and `LoadPackage` (**Reference: LoadPackage**) did not work correctly when one asked for a particular version of the package, via a version number starting with the character =, in the sense that a version with a larger version number was loaded if it was available. [Reported by Burkhard Höfling]
- The generator names constructed by `AlgebraByStructureConstants` (**Reference: AlgebraByStructureConstants**) were nonsense.
- The undocumented function (but recently advertised on gap-dev) `COPY_LIST_ENTRIES` did not handle overlapping source and destination areas correctly in some cases.

- The elements in a free magma ring have the filter `IsAssociativeElement` (**Reference: IsAssociativeElement**) set whenever the elements in the underlying magma and in the coefficients ring have this filter set. [Reported by Randy Cone]
- The function `InstallValue` (**Reference: InstallValue**) must not be used for objects in the filter `IsFamily` because these objects are compared via `IsIdenticalObj` (**Reference: IsIdenticalObj**). [Reported by Max Neunhöffer]

Fixed bugs which could lead to crashes:

- Problem in composition series for permutation groups for non-Frobenius groups with regular point stabilizer.
- After lots of computations with compressed GF(2) vectors GAP occasionally crashed. The reason were three missing `CHANGED_BAG`s in `SemiEchelonPListGF2Vecs`. They were missing, because a garbage collection could be triggered during the computation such that newly created bags could become "old". It is not possible to provide test code because the error condition cannot easily be reproduced. [Reported by Klaus Lux]
- Minor bug that crashed GAP: The type of `IMPLICATIONS` could not be determined in a fresh session. [Reported by Marco Costantini]
- `Assert` (**Reference: Assert**) caused an infinite loop if called as the first line of a function called from another function.

Other fixed bugs:

- Wrong choice of prime in Dixon-Schneider if prime is bigger than group order (if group has large exponent).
- Groebner basis code ran into problems when comparing monomial orderings.
- When testing for conjugacy of a primitive group to an imprimitive group,GAP runs into an error in EARNS calculation. [Reported by John Jones]
- The centre of a magma is commonly defined to be the set of elements that commute and associate with all elements. The previous definition left out "associate" and caused problems with extending the functionality to nonassociative loops. [Reported by Petr Vojtechovsky]
- New kernel methods for taking the intersection and difference between sets of substantially different sizes give a big performance increase.
- The commands `IsNaturalSymmetricGroup` (**Reference: IsNaturalSymmetricGroup**) and `IsNaturalAlternatingGroup` (**Reference: IsNaturalAlternatingGroup**) are faster and should run much less often into inefficient tests.
- The perfect group library, see `Finite Perfect Groups` (**Reference: Finite Perfect Groups**), is split into several files which are loaded and unloaded to keep memory usage down. The global variable `PERFSELECT` is a blist which indicates which orders are currently loaded. An off-by-one error wrongly added the last order of the previous file into the list of valid orders when a new file was loaded. A subsequent access to this order raises an error.

- Up to now, the method installed for testing the membership of rationals in the field of rationals via `IsRat` (**Reference: IsRat**) was not called; instead a more general method was used that called `Conductor` (**Reference: Conductor (for a cyclotomic)**) and thus was much slower. Now the special method has been ranked up by changing the requirements in the method installation.

- Fixed a bug in `APPEND_VEC8BIT`, which was triggered in the following situation: Let `e` be the number of field elements stored in one byte. If a compressed 8bit-vector `v` had length not divisible by `e` and another compressed 8-bit vector `w` was appended, such that the sum of the lengths became divisible by `e`, then one 0 byte too much was written, which destroyed the `TNUM` of the next GAP object in memory. [Reported by Klaus Lux]

- `PermutationCycle` (**Reference: PermutationCycle**) returned `fail` if the cycle was not a contiguous subset of the specified domain. [Reported by Luc Teirlinck]

- Now `Inverse` (**Reference: Inverse**) correctly returns `fail` for zeros in finite fields (and does no longer enter a break loop).

- Up to now, `CharacterDegrees` (**Reference: CharacterDegrees**) ignored the attribute `Irr` (**Reference: Irr**) if the argument was a group that knew that it was solvable.

- The function `Debug` now prints a meaningful message if the user tries to debug an operation. Also, the help file for `vi` is now available in the case of several GAP root directories.

- It is no longer possible to create corrupt objects via ranges of length $>2^{28}$, resp. $>2^{60}$ (depending on the architecture). The limitation concerning the arguments of ranges is documented. [Reported by Stefan Kohl]

- Now `IsElementaryAbelian` (**Reference: IsElementaryAbelian**) and `ClassPositionsOfMinimalNormalSubgroups` (**Reference: ClassPositionsOfMinimalNormalSubgroups**) are available for ordinary character tables. Now the operation `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) is an attribute, and there is another new attribute `SourceOfIsoclinicTable` (**Reference: SourceOfIsoclinicTable**) that points back to the original table; this is used for computing the Brauer tables of those tables in the character table library that are computed using `CharacterTableIsoclinic`. Now `ClassPositionsOfDerivedSubgroup` (**Reference: ClassPositionsOfDerivedSubgroup**) avoids calling `Irr` (**Reference: Irr**), since `LinearCharacters` (**Reference: LinearCharacters**) is sufficient. Now `ClassPositionsOfElementaryAbelianSeries` (**Reference: ClassPositionsOfElementaryAbelianSeries**) works also for the table of the trivial group. Restrictions of character objects know that they are characters.

  A few formulations in the documentation concerning character tables have been improved slightly.

- Up to now, `IsPGroup` (**Reference: IsPGroup**) has rarely been set. Now many basic operations such as `SylowSubgroup` (**Reference: SylowSubgroup**) set this attribute on the returned result.

- Computing an enumerator for a semigroup required too much time because it used all elements instead of the given generators. [Reported by Manuel Delgado]

- Avoid potential error message when working with automorphism groups.

- Fixed wrong page references in manual indices.
- Make `MutableCopyMat` an operation and install the former function which does call `List` (**Reference: Lists**) with `ShallowCopy` (**Reference: ShallowCopy**) the default method for lists. Also use this in a few appropriate places.
- An old DEC compiler doesn't like C preprocessor directives that are preceded by whitespace. Removed such whitespace. [Reported by Chris Wensley]

New or improved functionality:

- The primitive groups library has been extended to degree 2499.
- New operation `Remove` (**Reference: Remove**) and extended functionality of `Add` (**Reference: Add**) with an optional argument giving the position of the insertion. They are based on an efficient kernel function `COPY_LIST_ENTRIES`.
- Added fast kernel implementation of Tarjan's algorithm for strongly connected components of a directed graph.
- Now `IsProbablyPrimeInt` (**Reference: IsProbablyPrimeInt**) can be used with larger numbers. (Made internal function `TraceModQF` non-recursive.)
- A new operation `PadicValuation` (**Reference: PadicValuation**) and a corresponding method for rationals.
- A new operation `PartialFactorization` (**Reference: PartialFactorization**) has been added, and a corresponding method for integers has been installed. This method allows one to specify the amount of work to be spent on looking for factors.
- The generators of full s. c. algebras can now be accessed with the dot operator. [Reported by Marcus Bishop]
- New Conway polynomials computed by Kate Minola, John Bray, Richard Parker.
- A new attribute `EpimorphismFromFreeGroup` (**Reference: EpimorphismFromFreeGroup**). The code has been written by Alexander Hulpke.
- The functions `Lambda` (**Reference: Lambda**), `Phi` (**Reference: Phi**), `Sigma` (**Reference: Sigma**), and `Tau` (**Reference: Tau**) have been turned into operations, to admit the installation of methods for arguments other than integers.
- Up to now, one could assign only lists with `InstallFlushableValue` (**Reference: InstallFlushableValue**). Now also records are admitted.
- `InstallMethod` (**Reference: InstallMethod**) now admits entering a list of strings instead of a list of required filters. Each such string must evaluate to a filter when used as the argument of `EvalString` (**Reference: EvalString**). The advantage of this variant is that these strings are used to compose an info string (which is shown by `ApplicableMethod`) that reflects exactly the required filters.

- In test files that are read with `ReadTest` (**Reference: ReadTest**), the assertion level is set to 2 between `START_TEST` and `STOP_TEST`. This may result in runtimes for the tests that are substantially longer than the usual runtimes with default assertion level 0. In particular this is the reason why some of the standard test files require more time in GAP 4.4.6 than in GAP 4.4.5.
- Some very basic functionality for floats.

5.6 GAP 4.4 Update 7 (March 2006)

New or improved functionality:

- The `Display` (**Reference: Display**) functionality for character tables has been extended by addition of an option to show power maps and centralizer orders in a format similar to that used in the ATLAS. Furthermore the options handling is now hierarchical, in order to admit more flexible overloading.
- For the function `LowIndexSubgroupsFpGroup` (**Reference: LowIndexSubgroupsFpGroup**), there is now an iterator variant `LowIndexSubgroupsFpGroupIterator`. [Suggested (and based on code contributed) by Michael Hartley]
- Semigroup functionality in GAP has been improved and extended. Green's relations are now stored differently, making the system more amenable to new methods for computing these relations in special cases. It is now possible to calculate Green's classes etc. without computing the entire semigroup or necessarily loading the package MONOID. Furthermore, the Froidure-Pin algorithm has now been implemented in GAP.
- Functionality for creating free products of any list of groups for which a finite presentation can be determined had been added. This function returns a finitely presented group. This functionality includes the `Embedding` operation. As an application of this new code a specialized direct product operation has been added for finitely presented groups which returns a finitely presented group. This application includes `Embedding` and `Projection` functionality.
- Some new Straight Line Program (SLP) functionality has been added. The new functions take given SLPs and create new ones by restricting to a subset of the results, or to an intermediate result or by calculating the product of the results of two SLPs.
- New code has been added to allow group elements with memory; that is, they store automatically how they were derived from some given set of generators. Note that there is not yet documentation for this functionality, but some packages already use it.
- New code has been added to handle matrices and vectors in such a way that they do not change their representation in a generic manner.
- The `Irr` (**Reference: Irr**) method for p-solvable p-modular Brauer tables now keeps the order of the irreducibles in the ordinary table.
- GAP can now handle any finite field for which the Conway polynomial is known or can be computed.

- New Conway polynomials provided by John Bray and Kate Minola have been added.
- The `ReadTest` (**Reference: ReadTest**) methods for strings (filenames) and streams now automatically set the screen width (see `SizeScreen` (**Reference: SizeScreen**)) to 80 before the tests, and reset it afterwards.
- Now a few more checks are done during the `configure` phase of compiling for future use of some I/O functions of the C-library in a package. Also the path to the GAP binaries for the GAP compiler is now handled via autoconf. Finally, now `autoconf` version 2.59 is used.

Fixed bugs which could produce wrong results:

- Some technical errors in the functions for compressed vectors and matrices which could lead to corruption of internal data structures and so to crashes or conceivably to wrong results. [Reported by Roman Schmied]
- A potential problem in the generic method for the undocumented operation `DirectFactorsOfGroup`: It was silently assumed that `NormalSubgroups` (**Reference: NormalSubgroups**) delivers the trivial subgroup as first and the whole group as last entry of the resulting list.
- The code for sublists of compressed vectors created by `vec{range}` may write one byte beyond the space allocated for the new vector, overwriting part of the next object in the workspace. Thanks to Jack Schmidt for narrowing down the problem.
- Given a class function object of value zero, an `Arithmetic Operations for Class Functions` (**Reference: Arithmetic Operations for Class Functions**) method for a class function erroneously did not return `fail`. [Reported by Jack Schmidt]
- The `Arithmetic Operations for Class Functions` (**Reference: Arithmetic Operations for Class Functions**) method for a class function erroneously returned a finite number if one of the values was nonreal, not a cyclotomic integer, and had norm 1.
- Two missing perfect groups were added, and the permutation degree lowered on the perfect groups with the largest degrees. [Reported by Jack Schmidt]
- When a character table was displayed with `Printing Character Tables` (**Reference: Printing Character Tables**), the centralizer order displayed for the first class shown was not correct if it did not involve all prime divisors of the group. [Reported by Jack Schmidt]
- The first argument of the function `VectorSpace` (**Reference: VectorSpace**) must be a field. This is checked from now on. [Reported by Laurent Bartholdi]
- Up to now, it was possible to create a group object from a semigroup of cyclotomics using `AsGroup` (**Reference: AsGroup**), although groups of cyclotomics are not admissible. [Reported by Alexander Konovalov]
- The documentation of `CharacteristicPolynomial(F,mat)` was ambiguous if `FieldOfMatrix(mat) <= F < DefaultFieldOfMatrix(mat)`. In particular, the result was representation dependent. This was fixed by introducing a second field which specifies the vector space which mat acts upon. [Reported by Jack Schmidt]

- `AssociatedReesMatrixSemigroupOfDClass` produced an incorrect sandwich matrix for the semigroup created. This matrix is an attribute set when creating the Rees matrix semigroup but is not used for creating the semigroup. The incorrect result was returned when `SandwichMatrix` was called. [Reported by Nelson Silva and Joao Araujo]
- The literal `"compiled"` was given an incorrect length. The kernel was then unable to find compiled library code as the search path was incorrect. Also the documentation example had an error in the path used to invoke the `gac` compiler.
- The twisting group in a generic wreath product might have had intransitive action. [Reported by Laurent Bartholdi]
- There was an arithmetic bug in the polynomial reduction code.

Fixed bugs which could lead to crashes:

- Bug 1 in the list of fixed bugs which could lead to wrong results could also potentially lead to crashes.

Other fixed bugs:

- The matrices of invariant forms stored as values of the attributes `InvariantBilinearForm` (**Reference: InvariantBilinearForm**), `InvariantQuadraticForm` (**Reference: InvariantQuadraticForm**), and `InvariantSesquilinearForm` (**Reference: InvariantSesquilinearForm**), for matrix groups over finite fields, are now in the (compressed) format returned by `ImmutableMatrix` (**Reference: ImmutableMatrix**).
- `String` now returns an immutable string, by making a copy before changing the argument.
- `permutation^0` and `permutation^1` were not handled with special code in the kernel, hence were very slow for big permutations. [Reported by Max Neunhöffer]
- Added code to cache the induced pcgs for an arbitrary parent pcgs. (This code was formerly part of the CRISP package.)
- This fix consists of numerous changes to improve support for direct products, including: - new methods for `PcgsElementaryAbelianSeries`, `PcgsChiefSeries`, `ExponentsOfPcElement`, `DepthOfPcElement` for direct products - fixed `EnumeratorOfPcgs` to test for membership first - new methods for membership test in groups which have an induced pcgs - added `GroupOfPcgs` attribute to pcgs in various methods - fixed declarations of `PcgsElementaryAbelianSeries`, `PcgsChiefSeries` (the declared argument was a pcgs, not a group) [Reported by Roman Schmied]
- Corrected a term ordering problem encountered by the basis construction code for finite dimensional vector spaces of multivariate rational functions. [Reported by Jan Draisma]
- When the factor of a finite dimensional group ring by an ideal was formed, a method intended for free algebras modulo relations was used, and the returned factor algebra could be used for (almost) nothing. [Reported by Heiko Dietrich]
- Up to now, `PowerMap` (**Reference: PowerMap**) ran into an error when one asked for the n-th power map where n was not a small integer. This happened in some GAP library functions if the exponent of the character table in question was not a small integer.

- Up to now, the test whether a finite field element was contained in a group of finite field elements ran into an error if the element was not in the field generated by the group elements. [Reported by Heiko Dietrich]
- Conjugacy classes of natural (special) linear groups are now always returned with trivial class first.
- Up to now, it could happen that `CheckFixedPoints` (**Reference: CheckFixedPoints**) reduced an entry in its second argument to a list containing only one integer but did not replace the list by that integer; according to the conventions, this replacement should be done.
- The functions `PrintTo` and `AppendTo` did not work correctly for streams. [Reported by Marco Costantini]
- The function `Basis` did not return a value when it was called with the argument `Rationals`. [Reported by Klaus Lux]
- For certain matrix groups, the function `StructureDescription` raised an error message. The reason for this was that a trivial method for `IsGeneralLinearGroup` for matrix groups in `lib/grpmat.gi` which is ranked higher than the nontrivial method for generic groups in `lib/grpnames.gi` called the operation `IsNaturalGL`, for which there was no nontrivial method available. [Reported by Nilo de Roock]
- Action on sets of length 1 was not correctly handled. [Reported by Mathieu Dutour]
- Now `WriteByte` admits writing zero characters to all streams. [Reported by Marco Costantini]
- The conjugacy test for subgroups tests for elementary abelian regular normal subgroup (EARNS) conjugacy. The fix will catch this in the case that the second group has no EARNS. [Reported by Andrew Johnson]
- So far, the UNIX installation didn't result in a correct gap.sh if the installation path contained space characters. Now it should handle this case correctly, as well as other unusual characters in path names (except for double quotes).

5.7 GAP 4.4 Update 8 (September 2006)

New or improved functionality:

- A function `Positions` (**Reference: Positions**) with underlying operation `PositionsOp`, which returns the list of all positions at which a given object appears in a given list.
- `LogFFE` (**Reference: LogFFE**) now returns `fail` when the element is not a power of the base.
- It is now allowed to continue long integers, strings or identifiers by ending a line with a backslash or with a backslash and carriage return character. So, files with GAP code and DOS/Windows-style line breaks are now valid input on all architectures.
- The command line for starting the session and the system environment are now available in `GAPInfo.SystemCommandLine` and `GAPInfo.SystemEnvironment`.

- Names of all bound global variables and all component names are available on GAP level.
- Added a few new Conway polynomials computed by Kate Minola and John Bray.
- There is a new concept of *random sources*, see `IsRandomSource` (**Reference: IsRandomSource**), which provides random number generators which are independent of each other. There is kernel code for the Mersenne twister random number generator (based on the code by Makoto Matsumoto distributed at `http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html`). It provides fast 32-bit pseudorandom integers with a period of length $2^{19937} - 1$ and a 623-dimensional equidistribution. The library methods for random elements of lists and for random (long) integers are using the Mersenne twister now.
- In line editing mode (usual input mode without -n option) in lines starting with `gap>` , `>` or `brk>` this beginning part is immediately removed. This is a convenient feature that allows one to cut and paste input lines from other sessions or from manual examples into the current session.

Fixed bugs which could produce wrong results:

- The function `Decomposition` (**Reference: Decomposition**) returned coefficient vectors also in certain situations where in fact no decomposition exists. This happened only if the matrix entered as the first argument contained irrational values and a row in the matrix entered as the second argument did not respect the algebraic conjugacy relations between the columns of the first argument. So there was no problem for the usual cases that the two matrices are integral or that they are lists of Brauer characters. [Reported by Jürgen Müller]
- PC group homomorphisms can claim a wrong kernel after composition. [Reported by Serge Bouc]
- The return value of `OctaveAlgebra` (**Reference: OctaveAlgebra**) had an inconsistent defining structure constants table for the case of coefficients fields not containing the integer zero. [Reported by Gábor Nagy]
- The manual guarantees that a conjugator automorphism has a conjugating element in the group if possible. This was not guaranteed.
- `StabChain` (**Reference: StabChain (for a group (and a record))**) for symmetric groups gave a wrong result if fixed points were prescribed for base.
- Contrary to what is documented the function `POW_OBJ_INT` returned an immutable result for `POW_OBJ_INT(m,1)` for a mutable object `m`. This is triggered by the code `m^1`.
- `PseudoRandom` (**Reference: PseudoRandom**) for a group had a problem if the group had lots of equal generators. The produced elements were extremely poorly distributed in that case. This is now fixed for the case that elements of the group can easily be sorted.
- Fixed the bug that the type of a boolean list (see `More about Boolean Lists` (**Reference: More about Boolean Lists**)) was computed wrongly: The type previously had `IS_PLIST_REP` instead of `IS_BLIST_REP` in its filter list.

- `Orbits` (**Reference: Orbits**) did not respect a special `PositionCanonical` (**Reference: PositionCanonical**) method for right transversals. [Reported by Steve Costenoble]
- Wrong results for `GcdInt` (**Reference: GcdInt**) for some arguments on 64 bit systems only. [Reported by Robert Morse]
- When prescribing a subgroup to be included, the low index algorithm for fp groups sometimes returned subgroups which are in fact conjugate. (No subgroups are missing.) [Reported by Ignaz Soroko]

Fixed bugs which could lead to crashes:

- The command line option `-x` allowed arguments > 256 which can then result in internal buffers overflowing. Now bigger numbers in the argument are equivalent to `-x 256`. [Reported by Michael Hartley]

Other fixed bugs:

- Two special methods for the operation `CompositionMapping2` (**Reference: CompositionMapping2**) were not correct, such that composing (and multiplying) certain group homomorphisms did not work. [Reported by Peter Mayr]
- In the definition of `FrobeniusCharacterValue` (**Reference: FrobeniusCharacterValue**), it had been stated erroneously that the value must lie in the field of p^n-th roots of unity; the correct condition is that the value must lie in the field of $(p^n - 1)$-th roots of unity. [Reported by Jack Schmidt]
- The function `DirectProduct` (**Reference: DirectProduct**) failed when one of the factors was known to be infinite.
- For a linear action homomorphism `PreImageElm` was very slow because there was no good method to check for injectivity, which is needed for nearly all good methods for `PreImageElm`. This change adds such a new method for `IsInjective`. [Reported by Akos Seress]
- Rare errors in the complement routine for permutation groups.
- Blocks code now uses jellyfish-style random elements to avoid bad Schreier trees.
- A method for `IsPolycyclicGroup` (**Reference: IsPolycyclicGroup**) has been added. Such a method was missing so far.
- Corrected `EpimorphismSchurCover` (**Reference: EpimorphismSchurCover**) to handle the trivial group correctly. Added new methods that follow immediately from computing the Schur Cover of a group. The attribute `Epicentre` (**Reference: Epicentre**), the operations `NonabelianExteriorSquare` (**Reference: NonabelianExteriorSquare**) and `EpimorphismNonabelianExteriorSquare` (**Reference: EpimorphismNonabelianExteriorSquare**), and the property `IsCentralFactor` (**Reference: IsCentralFactor**) are added to the library with documentation and references.
- Display the correct expression in a call stack trace if an operation was called somewhere up due to the evaluation of a unary or binary operation.

- Made `StripMemory` an operation rather than a global function. Added `ForgetMemory` operation.
- Adjust things slightly to make later conversion to new vectors/matrices easier. Nothing of this should be visible.
- Corrected some details in the documentation of the GAP language. [Reported by Alexander Konovalov]
- Now `PositionSorted` (**Reference: PositionSorted**) is much faster on long mutable plain lists. (The former operation is substituted by a function and a new operation `PositionSortedOp`.) [Reported by Silviu Radu]
- Now it is possible to switch repeated warnings off when working with iterative polynomial rings.

5.8 GAP 4.4 Update 9 (November 2006)

Fixed bugs which could produce wrong results:

- The methods of `ReadByte` (**Reference: ReadByte**) for reading from files or terminals returned wrong results for characters in the range `[128..255]`. [Reported by Yevgen Muntyan]

Other fixed bugs:

- A method for the operation `PseudoRandom` (**Reference: PseudoRandom**) did not succeed.
- A fix for `Orbits` with a set of points as a seed.
- Added a generic method such that `Positions` (**Reference: Positions**) works with all types of lists.
- Fixed a problem in choosing the prime in the Dixon-Schneider algorithm. [Reported by Toshio Sumi]

New or improved functionality:

- `ReducedOrdinary` was used in the manual, but was not documented, being a synonym for the documented `ReducedCharacters`. Changed manual examples to use the latter form. [Reported by Vahid Dabbaghian]

5.9 GAP 4.4 Update 10 (October 2007)

New or improved functionality:

- Files in the `cnf` directory of the GAP distribution are now archived as binary files. Now GAP can be installed with UNIX or with WINDOWS style line breaks on any system and should work without problems.
- Since large finite fields are available, some restrictions in the code for computing irreducible modules over finite fields are no longer necessary. (They had been introduced in order to give better error messages.)

- Made PositionSublist faster in case the search string does not contain repetitive patterns.
- The function `MakeImmutable` now returns its argument.
- Dynamically loaded modules now work on Mac OS X. As a consequence, this allows to work with the Browse, EDIM and IO packages on Mac OS X.
- Introduced `ViewObj` and `PrintObj` methods for algebraic number fields. Made them applicable to `AlgebraicExtension` by adding the property `IsNumberField` in the infinite field case.
- The function `CharacterTableRegular` (**Reference: CharacterTableRegular**) is documented now.
- The function `ScalarProduct` (**Reference: ScalarProduct (for characters)**) now accepts also Brauer characters as arguments.
- The function `QuaternionAlgebra` (**Reference: QuaternionAlgebra**) now accepts also a list of field elements instead of a field. Also, now the comparison of return values (w.r.t. equality, containment) yields `true` if the parameters coincide and the ground fields fit.
- The function `RemoveCharacters` (**Reference: RemoveCharacters**) is now documented.
- Lists in GAP sometimes occupy memory for possible additional entries. Now plain lists and strings read by GAP and the lists returned by `List` (**Reference: Lists**) only occupy the memory they really need. For more details see the documentation of the new function `EmptyPlist` (**Reference: EmptyPlist**).
- There are some new Conway polynomials in characteristic 2 and 3 provided by Kate Minola.
- A new operation `MemoryUsage` determines the memory usage in bytes of an object and all its subobjects. It does not consider families and types but handles arbitrary self-referential structures of objects.

Fixed bugs which could produce wrong results:

- When forming the semidirect product of a matrix group with a vector space over a non-prime field the embedding of the vector space gave a wrong result. [Reported by anvita21]
- DefaultRing failed for constant polynomials over nonprime fields. [Reported by Stefan Kohl]
- The method in ffeconway.gi that gets coefficients WRT to the canonical basis of the field from the representation is only correct if the basis is over the prime field. Added a TryNextMethod if this is not the case. [Reported by Alla Detinko]
- Creating a large ($>2^{16}$) field over a non-prime subfield went completely wrong. [Reported by Jack Schmidt, from Alla Detinko]
- A method for Coefficients for Conway polynomial FFEs didn't check that the basis provided was the canonical basis of the RIGHT field. [Reported by Bettina Eick]
- An elementary abelian series was calculated wrongly. [Reported by N. Sieben]
- Orbits on sets of transformations failed.

- Wrong methods for `GeneratorsOfRing` (**Reference: GeneratorsOfRing**) and `GeneratorsOfRingWithOne` (**Reference: GeneratorsOfRingWithOne**) have been removed. These methods were based on the assumption that one can obtain a set of ring generators by taking the union of a known set of field generators, the set of the inverses of these field generators and `{1}`.
- The name of a group of order 117600 and degree 50 was incorrect in the `Primitive Permutation Groups` (**Reference: Primitive Permutation Groups**) Primitive Permutation Groups library. In particular, a group was wrongly labelled as PGL(2, 49).
- There was a possible error in `SubgroupsSolvableGroup` when computing subgroups within a subgroup.
- An error in 2-Cohomology computation for pc groups was fixed.
- `IsConjugate` used normality in a wrong supergroup

Fixed bugs which could lead to crashes:

- GAP crashed when the `PATH` environment variable was not set. [Reported by Robert F. Morse]
- GAP could crash when started with option `-x 1`. Now the number of columns is initialized with at least 2. [Reported by Robert F. Morse]
- After loading a saved workspace GAP crashed when one tried to slice a compressed vector over a field with 2 < q <= 256 elements, which had already existed in the saved workspace. [Reported by Laurent Bartholdi]
- `FFECONWAY.WriteOverSmallestCommonField` tripped up when the common field is smaller than the field over which some of the vector elements are written, because it did a test based on the degree of the element, not the field it is written over. [Reported by Thomas Breuer]
- Fixed the following error: When an FFE in the Conway polynomial representation actually lied in a field that is handled in the internal representation (eg $GF(3)$) and you tried to write it over a bigger field that is ALSO handled internally (eg $GF(9)$) you got an element written over the larger field, but in the Conway polynomial representation, which is forbidden. [Reported by Jack Schmidt]
- Attempting to compress a vector containing elements of a small finite field represented as elements of a bigger (external) field caused a segfault. [Reported by Edmund Robertson]
- GAP crashed when `BlistList` was called with a range and a list containing large integers or non-integers. [Reported by Laurent Bartholdi]
- GAP no longer crashes when `OnTuples` is called with a list that contains holes. [Reported by Thomas Breuer]

Other fixed bugs:

- `Socle` for the trivial group could produce an error message.
- `DirectoryContents` (**Reference: DirectoryContents**) ran into an error for immutable strings without trailing slash as argument. [Reported by Thomas Breuer]

- The functions `IsInjective` (**Reference: IsInjective**) and `IsSingleValued` (**Reference: IsSingleValued**) did not work for general linear mappings with trivial (pre)image. [Reported by Alper Odabas]
- Creating an enumerator for a prime field with more than 65536 elements ran into an infinite recursion. [Reported by Akos Seress]
- The performance of `List`, `Filtered`, `Number`, `ForAll` and `ForAny` if applied to non-internally represented lists was improved. Also the performance of iterators for lists was slightly improved.
- Finite field elements now know that they can be sorted easily which improves performance in certain lookups.
- A method for `IsSubset` (**Reference: IsSubset**) was missing for the case that exactly one argument is an inhomogeneous list. [Reported by Laurent Bartholdi]
- Long integers in expressions are now printed (was not yet implemented). [Reported by Thomas Breuer]
- Fixed kernel function for printing records.
- New C library interfaces (e.g., to ncurses in the Browse package) need some more memory to be allocated with `malloc`. The default value of GAP `-a` option is now `2m>`.
- Avoid warnings about pointer types by newer gcc compilers.
- `IsBound(l[pos])` was failing for a large integer *pos* only when coded (e.g. in a loop or function body).
- `ZmodpZObj` is now a synonym for `ZmodnZObj` such that from now on such objects print in a way that can be read back into GAP.
- The outdated note that binary streams are not yet implemented has been removed.

5.10 GAP 4.4 Update 11 (December 2008)

Fixed bugs which could produce wrong results:

- `MemoryUsage` (**Reference: MemoryUsage**) on objects with no subobjects left them in the cache and thus reported 0 in subsequent calls to MemoryUsage for the same object. [Reported by Stefan Kohl]
- `Irr` (**Reference: Irr**) might be missing characters. [Reported by Angel del Rio]
- Up to now, it was allowed to call the function `FullMatrixAlgebraCentralizer` (**Reference: FullMatrixAlgebraCentralizer**) with a field and a list of matrices such that the entries of the matrices were not contained in the field; in this situation, the result did not fit to the documentation. Now the entries of the matrices are required to lie in the field, if not then an error is signaled.

- For those finite fields that are regarded as field extensions over non-prime fields (one can construct such fields with `AsField` (**Reference: AsField**)), the function `DefiningPolynomial` (**Reference: DefiningPolynomial**) erroneously returned a polynomial w.r.t. the extension of the prime field. [Reported by Stefan Kohl]

- Since the release of GAP 4.4.10, the return values of the function `QuaternionAlgebra` (**Reference: QuaternionAlgebra**) were not consistent w.r.t. the attribute `GeneratorsOfAlgebra` (**Reference: GeneratorsOfAlgebra**); the returned list could have length four or five. Now always the list of elements of the canonical basis is returned.

- `MonomialGrevlexOrdering` (**Reference: MonomialGrevlexOrdering**) calculated a wrong ordering in certain cases. [Reported by Paul Smith]

- The (GAP kernel) method for the operation `IntersectSet` (**Reference: IntersectSet**) for ranges had two bugs, which could yield a result range with either too few or too many elements. As a consequence, for example the `Intersection` (**Reference: Intersection**) results for ranges could be wrong. [Reported by Matthew Fayers]

- Fixed a bug in the short-form display of elements of larger finite fields, a bug in some cross-field conversions and some inefficiencies and a missing method in the `LogFFE` (**Reference: LogFFE**) code. [Reported by Jia Huang]

- In rare cases `SmithNormalFormIntegerMatTransforms` (**Reference: SmithNormalFormIntegerMatTransforms**) returned a wrong normal form (the version without transforming matrices did not have this problem). This is fixed. [Reported by Alexander Hulpke]

- The variant of the function `StraightLineProgram` (**Reference: StraightLineProgram (for a list of lines (and the number of generators))**) that takes a string as its first argument returned wrong results if the last character of this string was a closing bracket.

- The code for central series in a permutation group used too tight a bound and thus falsely return a nilpotent permutation group as non-nilpotent.

Fixed bugs which could lead to crashes:

- Under certain circumstances the kernel code for position in blists would access a memory location just after the end of the blist. If this location was not accessible, a crash could result. This was corrected and the code was cleaned up. [Reported by Alexander Hulpke]

Other fixed bugs:

- The function `IsomorphismTypeInfoFiniteSimpleGroup` (**Reference: IsomorphismTypeInfoFiniteSimpleGroup**) can be called with a positive integer instead of a group, and then returns information about the simple group(s) of this order. (This feature is currently undocumented.) For the argument 1, however, it ran into an infinite loop.

- A lookup in an empty dictionary entered a break loop. Now returns `fail`. [Reported by Laurent Bartholdi]

- The c++ keyword `and` can no longer be used as a macro parameter in the kernel. [Reported by Paul Smith]

- The operation KernelOfMultiplicativeGeneralMapping (**Reference: KernelOfMultiplicativeGeneralMapping**) has methods designed to handle maps between permutation groups in a two-step approach, but did not reliably trigger the second step. This has now been fixed, preventing a slow infinite loop repeating the first step. This was normally only seen as part of a larger calculation.

- There were two methods for the operation Intersection2 (**Reference: Intersection2**) which have implicitly assumed that finiteness of a collection can always be decided. Now, these methods check for IsFinite (**Reference: IsFinite**) and CanComputeSize (**Reference: CanComputeSize**) prior to calling IsFinite (**Reference: IsFinite**).

- Made error message in case of corrupted help book information (manual.six file) shorter and more informative. [Reported by Alexander Hulpke]

- GAP cannot call methods with more than six arguments. Now the functions NewOperation (**Reference: NewOperation**), DeclareOperation (**Reference: DeclareOperation**), and InstallMethod (**Reference: InstallMethod**) signal an error if one attempts to declare an operation or to install a method with more than six arguments.

- Up to now, IsOne (**Reference: IsOne**) had a special method for general mappings, which was much worse than the generic method; this special method has now been removed.

- When printing elements of an algebraic extension parentheses around coefficients were missing. [Reported by Maxim Hendriks]

New or improved functionality:

- Make dynamic loading of modules possible on CYGWIN using a DLL based approach. Also move to using autoconf version 2.61.

- One can now call Basis (**Reference: Basis**), Iterator (**Reference: Iterator**) etc. with the return value of the function AlgebraicExtension (**Reference: AlgebraicExtension**).

- The function FrobeniusCharacterValue (**Reference: FrobeniusCharacterValue**) returned fail for results that require a finite field with more than 65536 elements. Meanwhile GAP can handle larger finite fields, so this restriction was removed. (It is still possible that FrobeniusCharacterValue (**Reference: FrobeniusCharacterValue**) returns fail.)

- Methods for testing membership in general linear groups and special linear groups over the integers have been added.

- Methods for String (**Reference: String**) and ViewString for full row modules have been added. Further, a default method for IsRowModule (**Reference: IsRowModule**) has been added, which returns false for objects which are not free left modules.

- A ViewString method for objects with name has been added.

- The method for View (**Reference: View**) for polynomial rings has been improved, and methods for String (**Reference: String**) and ViewString for polynomial rings have been added.

- Binomial (**Reference: Binomial**) now works with huge n.

- The function `InducedClassFunctionsByFusionMap` (**Reference: InducedClassFunctionsByFusionMap**) is now documented.
- The return values of the function `QuaternionAlgebra` (**Reference: QuaternionAlgebra**) now store that they are division rings (if optional parameters are given then of course ths depends on these parameters).

5.11 GAP 4.4 Update 12 (December 2008)

Fixed bugs which could lead to crashes:

- A bug whereby leaving an incomplete statement on a line (for instance typing while and then return) when prompt colouring was in use could lead to GAP crashing.

Other fixed bugs:

- A bug which made the command-line editor unusable in a 64-bit version of GAP on Mac OS X.

Chapter 6

Changes from Earlier Versions

6.1 Changes between GAP 4.3 and GAP 4.4

The main changes between GAP 4.3 and GAP 4.4 are:

6.1.1 Potentially Incompatible Changes

- The mechanism for the loading of Packages has changed to allow easier updates independent of main GAP releases. Packages require a file `PackageInfo.g` now. The new `PackageInfo.g` files are available for all packages with the new version of GAP (see **Example: PackageInfo.g for a GAP package**).
- `IsSimpleGroup` (**Reference: IsSimpleGroup**) returns false now for the trivial group.
- `PrimeBlocks` (**Reference: PrimeBlocks**): The output format has changed.
- Division rings (see `IsDivisionRing` (**Reference: IsDivisionRing**)) are now implemented as `IsRingWithOne` (**Reference: IsRingWithOne**).
- `DirectSumOfAlgebras` (**Reference: DirectSumOfAlgebras (for two algebras)**): p-th power maps are compatible with the input now.
- The print order for polynomials has been changed.

These changes are, in some respects, departures from our policy of maintaining upward compatibility of documented functions between releases. In the first case, we felt that the old behavior was sufficiently inconsistent, illogical, and impossible to document that we had no alternative but to change it. In the case of the package interface, the change was necessary to introduce new functionality. The planned and phased removal of a few unnecessary functions or synonyms is needed to avoid becoming buried in "legacy" interfaces, but we remain committed to our policy of maintaining upward compatibility whenever sensibly possible.

- Groebner Bases:

  Buchberger's algorithm to compute Groebner Bases has been implemented in GAP. (A. Hulpke)

- For large scale Groebner Basis computations there also is an interface to the Singular system available in the Singular package. (M. Costantini and W. de Graaf)

- New methods for factorizing polynomials over algebraic extensions of the rationals have been implemented in GAP. (A. Hulpke)
- For more functionality to compute with algebraic number fields there is an interface to the Kant system available in the Alnuth package. (B. Assmann and B. Eick)
- A new functionality to compute the minimal normal subgroups of a finite group, as well as its socle, has been installed. (B. Höfling)
- A fast method for recognizing whether a permutation group is symmetric or alternating is available now (A. Seress)
- A method for computing the Galois group of a rational polynomial is available again. (A. Hulpke)
- The algorithm for `BrauerCharacterValue` (**Reference: BrauerCharacterValue**) has been extended to the case where the splitting field is not supported in GAP. (T. Breuer)
- Brauer tables of direct products can now be constructed from the known Brauer tables of the direct factors. (T. Breuer)
- Basic support for vector spaces of rational functions and of uea elements is available now in GAP. (T. Breuer and W. de Graaf)
- Various new functions for computations with integer matrices are available, such as methods for computing normal forms of integer matrices as well as nullspaces or solutions systems of equations. (W. Nickel and F. Gähler)

6.1.2 New Packages

The following new Packages have been accepted.

- Alnuth: Algebraic Number Theory and an interface to the Kant system. By B. Assmann and B. Eick.
- LAGUNA: Computing with Lie Algebras and Units of Group Algebras. By V. Bovdi, A. Konovalov, R. Rossmanith, C. Schneider.
- NQ: The ANU Nilpotent Quotient Algorithm. By W. Nickel.
- KBMAG: Knuth-Bendix for Monoids and Groups. By D. Holt.
- Polycyclic: Computation with polycyclic groups. By B. Eick and W. Nickel.
- QuaGroup: Computing with Quantized Enveloping Algebras. By W. de Graaf.

6.1.3 Performance Enhancements

- The computation of irreducible representations and irreducible characters using the Baum-Clausen algorithm and the implementation of the Dixon-Schneider algorithm have been speeded up.

- The algorithm for `PossibleClassFusions` (**Reference: PossibleClassFusions**) has been changed: the efficiency is improved and a new criterion is used. The algorithm for `PossibleFusionsCharTableTom` (**Reference: PossibleFusionsCharTableTom**) has been speeded up. The method for `PrimeBlocks` (**Reference: PrimeBlocks**) has been improved following a suggestion of H. Pahlings.

- New improved methods for normalizer and subgroup conjugation in S_n have been installed and new improved methods for `IsNaturalSymmetricGroup` (**Reference: IsNaturalSymmetricGroup**) and `IsNaturalAlternatingGroup` (**Reference: IsNaturalAlternatingGroup**) have been implemented. These improve the available methods when groups of large degrees are given.

- The partition split method used in the permutation backtrack is now in the kernel. Transversal computations in large permutation groups are improved. Homomorphisms from free groups into permutation groups now give substantially shorter words for preimages.

- The membership test in `SP` (**Reference: Sp (for dimension and field size)**) and `SU` (**Reference: SU**) groups has been improved using the invariant forms underlying these groups.

- An improvement for the cyclic extension method for the computation of subgroup lattices has been implemented.

- A better method for `MinimalPolynomial` (**Reference: MinimalPolynomial**) for finite field matrices has been implemented.

- The display has changed and the arithmetic of multivariate polynomials has been improved.

- The `LogMod` (**Reference: LogMod**) function now uses Pollard's rho method combined with the Pohlig/Hellmann approach.

- Various functions for sets and lists have been improved following suggestions of L. Teirlinck. These include: `Sort` (**Reference: Sort**), `Sortex` (**Reference: Sortex**), `SortParallel` (**Reference: SortParallel**), `SortingPerm` (**Reference: SortingPerm**), `NrArrangements` (**Reference: NrArrangements**).

- The methods for `StructureConstantsTable` (**Reference: StructureConstantsTable**) and `GapInputSCTable` (**Reference: GapInputSCTable**) have been improved in the case of a known (anti-) symmetry following a suggestion of M. Costantini.

The improvements listed in this Section have been implemented by T. Breuer and A. Hulpke.

6.1.4 New Programming and User Features

- The 2GB limit for workspace size has been removed and version numbers for saved workspaces have been introduced. (S. Linton and B. Höfling)

- The limit on the total number of types created in a session has been removed. (S. Linton)

- There is a new mechanism for loading packages available. Packages need a file `PackageInfo.g` now. (T. Breuer and F. Lübeck; see **Example: PackageInfo.g for a GAP package**).

Finally, as always, a number of bugs have been fixed. This release thus incorporates the contents of all the bug fixes which were released for GAP 4.3. It also fixes a number of bugs discovered since the last bug fix.

6.2 Earlier Changes

The most important changes between GAP 4.2 and GAP 4.3 were:

- The performance of several routines has been substantially improved.
- The functionality in the areas of finitely presented groups, Schur covers and the calculation of representations has been extended.
- The data libraries of transitive groups, finite integral matrix groups, character tables and tables of marks have been extended.
- The Windows installation has been simplified for the case where you are installing GAP in its standard location.
- Many bugs have been fixed.

The most important changes between GAP 4.1 and GAP 4.2 were:

- A much extended and improved library of small groups as well as associated `IdGroup` (**Reference: IdGroup**) routines.
- The primitive groups library has been made more independent of the rest of GAP, some errors were corrected.
- New (and often much faster) infrastructure for orbit computation, based on a general "dictionary" abstraction.
- New functionality for dealing with representations of algebras, and in particular for semisimple Lie algebras.
- New functionality for binary relations on arbitrary sets, magmas and semigroups.
- Bidirectional streams, allowing an external process to be started and then controlled "interactively" by GAP
- A prototype implementation of algorithms using general subgroup chains.
- Changes in the behavior of vectors over small finite fields.
- A fifth book "New features for Developers" has been added to the GAP manual.
- Numerous bug fixes and performance improvements

The changes between the final release of GAP 3 (version 3.4.4) and GAP 4 are wide-ranging. The general philosophy of the changes is two-fold. Firstly, many assumptions in the design of GAP 3 revealed its authors' primary interest in group theory, and indeed in finite group theory. Although much of the GAP 4 library is concerned with groups, the basic design now allows extension to other algebraic structures, as witnessed by the inclusion of substantial bodies of algorithms for computation with semigroups and Lie algebras. Secondly, as the scale of the system, and the number of people using and contributing to it has grown, some aspects of the underlying system have proved to be restricting, and these have been improved as part of comprehensive re-engineering of the system. This has included the new method selection system, which underpins the library, and a new, much more flexible, GAP package interface.

Details of these changes can be found in the document "Migrating to GAP 4" available at the GAP website, see http://www.gap-system.org/Gap3/migratedoc.pdf.

It is perhaps worth mentioning a few points here.

Firstly, much remains unchanged, from the perspective of the mathematical user:

- The syntax of that part of the GAP language that most users need for investigating mathematical problems.
- The great majority of function names.
- Data libraries and the access to them.

A number of visible aspects have changed:

- Some function names that need finer specifications now that there are more structures available in GAP.
- The access to information already obtained about a mathematical structure. In GAP 3 such information about a group could be looked up by directly inspecting the group record, whereas in GAP 4 functions must be used to access such information.

Behind the scenes, much has changed:

- A new kernel, with improvements in memory management and in the language interpreter, as well as new features such as saving of workspaces and the possibility of compilation of GAP code into C.
- A new structure to the library, based upon a new type and method selection system, which is able to support a broader range of algebraic computation and to make the structure of the library simpler and more modular.
- New and faster algorithms in many mathematical areas.
- Data structures and algorithms for new mathematical objects, such as algebras and semigroups.
- A new and more flexible structure for the GAP installation and documentation, which means, for example, that a GAP package and its documentation can be installed and be fully usable without any changes to the GAP system.

Very few features of GAP 3 are not yet available in GAP 4.

- Not all of the GAP 3 packages have yet been converted for use with GAP 4.
- The library of crystallographic groups which was present in GAP 3 is now part of a GAP 4 package CrystCat by V. Felsch and F. Gähler.

Index

GAP binary distributions, 26
GAP compiler (no longer recommended), 26
GAP 3 compatibility mode (withdrawn), 26
GMP support, 22
MathJax support, 28
readline support, 27

Bugfixes and packages archives (withdrawn), 26

Completion files (withdrawn), 26

Data libraries, 23

Floats, 23

Namespaces, 29

Packages under Windows, 33
Packages, new, 10
Packages, upgraded, 10

`tools` archive, 26

User interface customisation, 27