Summary of New Features in Magma V2.5

July 7, 1999

1 Introduction

This document provides a terse summary of the new features installed in Magma for release version V2.5 (July 7, 1999). Previous releases of Magma are: V2.4 (December 3, 1998), V2.3 (January 30, 1998), V2.20 (April 18, 1997), V2.10 (October 14, 1996), V2.01 (June 21, 1996), and V1.30 (March 5, 1996); release notes for these versions are found in the document `relprev.dvi`.

2 Summary

- The `kbmag` package of Derek Holt has been installed. This has as its basis a powerful Knuth-Bendix algorithm for monoids. Using this as a foundation, three new categories have been installed: `MonRWS` – monoids defined by finite confluent rewrite systems; `GrpRWS` – groups defined by finite confluent rewrite systems; and `GrpATC` – short-lex automatic groups.

- The Plesken soluble quotient algorithm has been implemented in Magma by Herbert Brückner. This is the first implementation capable of computing soluble quotients without the user needing to supply any information about the quotient.

- The ordinary irreducible representations of a finite soluble group may now be constructed. Previously, only the modular irreducible representations were available.

- A basic module for Coxeter groups implemented by Don Taylor has been installed.

- General algebraic function fields are available for the first time. These facilities are provided by the KANT group. A general function field is a finite degree extension of a rational function field $K(x)$. In Magma V2.5, the field K may be taken to be either a finite field, $\mathbf{Q}$ or a number field. Operations supported include the arithmetic of algebraic functions, genus, maximal orders, arithmetic of fractional ideals, fundamental unit computation (global function fields) and subfields.

- The first stage of a general package for working with plane curves is included in V2.5. Its main features are flexible tools for translating between affine and projective curves, the calculation of geometric genus of any plane curve and the explicit manipulation of divisors on cubics in Weierstraß form.

- Also forming part of the algebraic geometry machinery is a module for working with schemes defined by polynomial equations in affine or projective space. Such tools

include Gröbner basis computation, dimension and image of maps, and linear algebra calculations formalized as linear systems on projective space. Maps between spaces may also be constructed and studied.

- Magma V2.5 contains an efficient implementation of the Schoof-Elkies-Atkin algorithm with Lercier's extension to base fields of characteristic 2. Calculations are performed in the smallest field over which the curve is defined, and the result is lifted to the original field.

3 Removals and Changes

– The function `ElementaryDivisors` has been fixed to include any ones which are in the Smith normal form of the input matrix (so the length of the result is now always the rank of the input matrix).

– Descriptions of the functions `LargestClique` and `LargestIndependentSet` have been removed from the Magma Handbook. The functions will be removed from Magma in a subsequent release.

4 Language and System Features [HB 1–5]

New features:

– New functions `GetMemoryUsage`, `GetMaximumMemoryUsage` and `ResetMaximumMemoryUsage` for obtaining information about Magma's memory usage.

– New function `Getuid()` to return the user ID of the current Magma process (calls the system function `getuid()` in Unix).

– The expression `+`*specfile* may now be placed in a package specification file instead of a filename to indicate that all files in the named specification file *specfile* should be recursively attached.

– The help system browser now allows the command `grep` to search the full text of the help information for a given substring.

5 Semigroups

5.1 Monoids Defined by Rewrite Systems (New) [HB 17]

A new category comprising monoids with confluent rewriting systems (RWS monoids) has been introduced. The major algorithm provided is the Knuth-Bendix procedure as implemented in Derek Holt's *kbmag* package.

– Construction of an RWS monoid from an fp-monoid using the Knuth-Bendix procedure. Orderings supported include: *RT-recursive*, *recursive*, *ShortLex*, *WT-ShortLex* and *Wreath*.

– Test a rewrite system for confluence

– Reduction of a word to normal form

– Operations on words: Product, exponentiation, equality
– Test for a monoid being finite
– Enumeration of elements

6 Groups

6.1 Finitely Presented Groups [HB 20]

New features:

- The Plesken soluble quotient algorithm has been implemented for Magma by Herbert Brückner. This is the first implementation that will compute soluble quotients without having the user supply information about the quotient. It is capable of constructing very large finite quotients and also of detecting infinite soluble quotients.

6.2 Groups Defined by Rewrite Systems (New) [HB 21]

This is a category of finitely presented groups where the relations are interpreted as rewrite rules. If the group is defined by a confluent system of rewrite rules then we have a normal form for its elements and hence a solution to the word problem. A group belonging to this category is typically constructed by applying the Knuth-Bendix procedure. As in the case of monoids, Magma uses the Knuth-Bendix developed by Derek Holt as part of his package *kbmag*.

- Construction of an RWS group from an fp-group using the Knuth-Bendix procedure. Orderings supported include: *RT-recursive*, *recursive*, *ShortLex*, *WT-ShortLex* and *Wreath*
- Test a rewrite system for confluence
- Reduction of a word to normal form
- Operations on words: Product, exponentiation, inverse, equality
- Enumeration of elements
- Test for a group being finite

6.3 Automatic Groups (New) [HB 22]

This category corresponds to short-lex automatic groups. A group is represented by four automata: first and second word-difference machines, a word-acceptor, and a multiplier. These automata are constructed using the Knuth-Bendix procedure. This category is implemented by Derek Holt's package *kbmag*.

- Construction of an automatic group from an fp-group using the Knuth-Bendix procedure.
- Reduction of a word to normal form
- Product, exponentiation, inverse, equality of elements
- Enumeration of words without repetition
- Test for a group being finite
- Growth function for a group

6.4 Finite Soluble Groups [HB 25]

New features:

- Several functions are provided for constructing a polycyclic presentation for the maximal finite soluble quotient of an fp-group (Plesken-Brückner algorithm)
- The function `IrreducibleRepresentations` will compute the irreducible representations over a finite field, **Q**, or a cyclotomic number field.
- The function `AbsolutelyIrreducibleRepresentations` will compute the ordinary or modular absolutely irreducible representations.

Bug fixes:

- An error in the function `NormalSubgroups` where some normal subgroups were missed has been fixed.
- An error that showed up in rare situations in the subgroup echelonization code has been fixed. This would sometimes cause errors in the determination of conjugacy classes.

6.5 Coxeter Groups (New) [HB 28]

Finite Coxeter groups are implemented as a subclass of permutation groups so that they inherit all the operations for permutation groups as well as having many specialized functions. This module was implemented by Don Taylor. Frank Lübeck and the Chevie team provided helpful assistance.

- Cartan matrix corresponding to a given Dynkin diagram
- Construction of a Coxeter group from a root system or Cartan matrix
- Dynkin diagram of a Cartan matrix or Coxeter group
- Root system for a Coxeter group
- Element as a reduced word in the standard generators
- Element of maximal length
- Unique long (short) root of greatest height
- Short root of maximal height
- Reflections in Coxeter group
- Reflection subgroup
- Reduced representatives for cosets of the reflection subgroup
- Actions on roots and co-roots
- Coxeter group as a real reflection group

6.6 Groups of Lie Type (New) [HB 28]

- Killing form of the Cartan algebra associated with a given Weyl group
- Root elements
- Fundamental roots and their negatives of a simple Lie algebra of given type and rank
- Lie algebra of a Chevalley group as a structure constant algebra
- Adjoint action
- Graph automorphism of a Coxeter group
- Degree of a representation with specified weight
- The BN-pair for a group of Lie type

7 Rings

7.1 General Rings [HB 30]

New features:

- Mutation operators `?=` and `/:=` (when dividing by a unit) now work for all ring elements.

7.2 Finite Fields [HB 34]

Bug fixes:

- A bug in the iterator for certain finite field representations has been fixed (it sometimes produced duplicate elements).

7.3 Univariate Polynomial Rings [HB 35]

New features:

- New function `InverseMod/Modinv` for univariate polynomials to return the inverse of one polynomial modulo another.
- Coercion for rational function fields has been improved to avoid undesirable behaviour in various cases.
- Functions have been provided for creating polynomials belonging to the standard families of orthogonal polynomials. Included are Chebyshev polynomials of the first and second kind, Hermite polynomials, Legendre polynomials, Laguerre polynomials and Gegenbauer polynomials.
- Functions have been provided to construct instances of the Dickson polynomials of the first and second kind.

7.4 Multivariate Polynomial Rings [HB 36]

New features:

- The computation of the variety of an ideal over a (moderately small) finite field is now performed even when the dimension of the ideal is greater than zero.
- Computation of the (trivial) variety of the full polynomial ring is now allowed, and `VarietySizeOverAlgebraicClosure` now returns zero for the full polynomial ring.
- `SetVerbose("Groebner", 2);` will now cause all new computed polynomials to be printed fully in the Buchberger algorithm.

Bug fixes:

- A bug in multivariate factorization over the integer ring has been fixed.
- Error checking is now performed in `NormalForm` when a bad set or sequence is passed as the second argument.

7.5 Rational Function Fields, [HB 42]

New features:

- Coercion for rational function fields has been improved to avoid undesirable behaviour in various cases.

7.6 Global Function Fields and their Orders (New) [HB 43]

An algebraic function field is a finite degree extension of a rational function field $K(x)$. In Magma V2.5, the field K may be taken to be either a finite field, $\mathbf{Q}$ or a number field. An extension of $K(x)$, where K is finite field, is called a *global function field*. The development of this module is a joint project with the KANT group. Work is under way on developing algorithms for computing the divisor class group.

- Construction of a function field as an extension of $K(x)$
- Arithmetic with elements
- Trace and norm of an element
- Exact constant field
- Genus
- Finite and infinite equation orders
- Discriminant (of an order)
- Maximal order (finite or infinite)
- Construction of integral and fractional ideals
- Ideal arithmetic: sum, product, quotient, gcd, lcm
- Properties of ideals: Integral, prime, principal
- Valuation of an order element or ideal at a prime ideal
- Ramification index and residue class degree of a prime ideal
- Prime ideal decomposition of a fractional ideal
- Determination of places of degree one in global fields
- Determine whether the place at infinity place is tamely ramified
- Determination of places lying over a (finite) prime or the infinite place
- Basis reduction for finite orders (Pohst-Schörnig method) in global fields
- Independent units, fundamental units, regulator in global fields

7.7 Real and Complex Fields [HB 44]

Bug fixes:

- The function `DedekindEta` for real and complex fields has been fixed to give the correct result (which is consistent with the series function).

7.8 Formal Series [HB 46]

Bug fixes:

- Bug in the function `Eisenstein` fixed.
- Bug in the function `Evaluate` fixed.

8 Modules and Lattices

8.1 R-Modules [HB 49]

New features:

- New function `AHom` to compute the module of homomorphisms from one module to another which respect the action of the underlying matrix algebra (like `GHom` but the module need not be a G-module).

8.2 Chain Complexes (New) [HB 50]

New features:

- Creation of a complex from a list of A-modules
- Subcomplexes and quotient complexes
- Operations on complexes: Splice, shift, direct sum
- Exact extension
- Dual of a complex
- Homology groups of a complex
- Boundary maps
- Given two complexes construct the corresponding chain map
- Composition of chain maps
- Image, kernel and cokernel of a chain map
- Predicates for chain maps: Surjection, injection, isomorphism
- Injective resolution (for complexes over a basic algebra)
- Projective resolution (for complexes over a basic algebra)

8.3 Modules Hom(U, V) [HB 51]

Changes:

- The function `ElementaryDivisors` has been fixed to include any ones which are in the Smith normal form of the input matrix (so the length of the result is now always the rank of the input matrix).

9 Algebras

9.1 Matrix Algebras [HB 57]

New features:

- New modular algorithm for the multiplication of matrices over large prime finite fields.

Changes:

- The function `ElementaryDivisors` has been changed to include any ones occurring on the diagonal of the Smith normal form of the input matrix (so the length of the result is now always the rank of the input matrix).

9.2 Basic Algebras (New) [HB 59]

A basic algebra is a finite dimensional algebra A over a field, all of whose simple modules have dimension one. In the literature such an algebra is known as a "split" basic algebra. The type in Magma is optimized for the purposes of doing homological calculations. This module was written by Jon Carlson.

- Creation from a sequence of projective modules and a path tree for each module
- Creation of the basic algebra corresponding to the group algebra of a p-group over $GF(p)$.
- Arithmetic
- Extension and restriction of coefficient ring
- Tensor product
- Opposite algebra
- Injective hull
- Algebra as a right regular module over itself

10 Geometry

The algebraic geometry module includes machinery for studying general algebraic varieties and families of special curves (e.g. elliptic curves). The major categories include:

- Affine plane curves
- Projective plane curves
- Newton polygons
- Divisor groups of certain plane curves
- Schemes
- Elliptic curves

10.1 Algebraic Curves

Magma V2.5 includes the first stage of a general package for working with plane curves. These are interpreted as being the scheme defined by the vanishing of a polynomial defined on either an affine or projective space. Its main features are flexible tools for translating between affine and projective curves, the calculation of geometric genus of any plane curve and the explicit manipulation of divisors on cubics in Weierstraß form. Many more specialized will be included in future releases.

10.1.1 Plane Curves (New) [HB 62]

- Creation of affine and projective curves together with the ambient affine and projective planes
- Basic manoeuvres between affine and projective curves: projective closure, affine patches and so on
- Basic scheme-type functions: e.g., irreducibility
- Specialised data types for distinct classes of curves such as elliptic curves
- Linear systems of curves in the projective plane with assigned basepoints
- Implicitization of parametric curves
- Global invariants of curves, such as genus and dimension

10.1.2 Local Analysis of Curves (New) [HB 62]

- Calculation of tangent spaces and cones
- Identification of all singularities of a curve, together with the basic analysis
- Blowups, including weighted blowups, of points on curves
- Local intersections

10.1.3 Divisors and the Riemann-Roch Theorem (New) [HB 62]

The following functions apply to special classes of curves.

- Creation of divisors and arithmetic
- Compute the divisor of a function on a curve
- Determine whether a divisor is principal. If so find a function with the given divisor
- Normal forms for divisors

10.2 Newton Polygons (New) [HB 64]

- Construction of a newton polygon: Compact, infinite or including the origin
- Construction of newton polygons from different types of data: $f \in k[x, y]$, $f \in k\langle\langle x \rangle\rangle [y]$, a finite set of points; a finite set of faces (weighted dual vectors)
- Finding faces and vertices
- If polygon is derived from a polynomial f, finding restrictions of f to faces
- Locating a given point relative to a newton polygon
- Determining whether a given line supports a face
- Determining whether two polygons are the same

10.3 Schemes (New) [HB 63]

This module comprises general tools for working with schemes defined by polynomial equations in affine or projective space. Such tools include Gröbner basis computation, dimension and image of maps, and linear algebra calculations formalized as linear systems on projective space. Maps between spaces may also be constructed and studied.

- Creation of schemes by equations or implicit methods
- Basic algebra-theoretic analysis of schemes: reducibility, dimension
- Local geometric analysis: singularities, tangent spaces
- Construction of maps between spaces
- Coordinate manipulation functions, including birational transformations of the projective plane
- Calculation of pullbacks of schemes by maps
- Tools to enable the calculation of images of schemes by maps

10.4 Elliptic Curves [HB 65]

10.4.1 General Elliptic Curves

New features:

- New function `CremonaReference` to look up an elliptic curve over the rationals in the installed Cremona database.
- `EllipticCurve` and `EllipticCurves` extended to accept identifying strings (e.g. "101A") for curves in Cremona database.

10.4.2 Elliptic Curves over Finite Fields

Magma V2.5 contains an efficient implementation of the Schoof-Elkies-Atkin algorithm with Lercier's extension to base fields of characteristic 2. Calculations are performed in the smallest field over which the curve is defined, and the result is lifted to the original field.

New features for elliptic curves over finite fields:

- Schoof-Elkies-Atkin algorithm for counting points in the cases of large characteristic and characteristic 2.
- Use of Atkin's algorithm (as subset of the Schoof-Elkies-Atkin machinery) for point counting in small odd characteristic.
- New functions `IsProbablySupersingular`, `IsSupersingular`, and `IsOrdinary` for elliptic curves over finite fields, replacing `IsProvenSupersingular`.
- New function `FactoredOrder` for curves and points on curves.

Changes:

- Removal of function `Lift` from documentation. Functionality preserved with `BaseChange`, `BaseExtend`, and `ChangeRing`.
- `TraceOfFrobenius` as synonym for `Trace`.
- Added functions names `nTorsionSubgroup` and `pTorsionSubgroup`, synonymous with previous `mTorsionSubgroup`.
- Replacement of `IsProvenSupersingular` with new functions `IsProbablySupersingular`, `IsSupersingular`, and `IsOrdinary` for elliptic curves over finite fields.

11 Incidence Structures

11.1 Enumerative Combinatorics [HB 66]

New Features:

- A variant of the `DivisorSigma(i, n)` function has been included which allows the user to give n in factored form.
- The function `RestrictedPartitions` has been provided to construct the restricted partitions of an integer n, i.e., the partitions of n such that the parts belong to a designated set.
- A variant of the `Partitions` function has been provided which generates only the partitions having a specified number of parts.

11.2 Graphs [HB 67]

A new clique finding algorithm based on the use of recursion and dynamic programming has been implemented. In some circumstances, such at the case of large graphs with high density, this algorithm is more efficient than the existing branch and bound algorithm. The existing clique and independent set functions have been rewritten so to allow users to specify the choice of clique algorithm. We welcome any user feedback regarding the comparative performance of these two algorithms.
New features:

- The new function `MinimumDominatingSet` finds a minimum dominating set of an undirected graph.
- The new functions `OptimalVertexColouring` and `OptimalEdgeColouring` find one optimal vertex, respectively, one optimal edge colouring, in an undirected graph.
- The new function `ChromaticPolynomial` computes the chromatic polynomial for an undirected graph.
- Functions `AllCliques` and `AllIndependentSets` find all cliques and independent sets, respectively, of a given size. No choice of algorithm is permitted.

Changes:

- The function `Clique` now has an extra argument allowing the user to specify algorithm which is to be used. If no clique size is given, then `Clique` returns a maximum clique. If a clique size k is specified, then `Clique` returns a clique of size exactly k if the dynamic algorithm is chosen, or a maximal clique of size at least k if the branch and bound algorithm is chosen. The same comment applies to the function `IndependentSet`.
- The functions `CliqueNumber` and `IndependenceNumber` also admit an extra argument allowing the user to choose the algorithm used.
- Descriptions of the functions `LargestClique` and `LargestIndependentSet` have been removed from the Magma Handbook. The functions will be removed from Magma in a subsequent release.

11.3 Incidence Structures and Designs [HB 68]

Changes:

– A new algorithm has been implemented to find a resolution in an incidence structure. The function `IsResolvable` has an extra argument allowing to choose the algorithm which is to be used. The previously existing algorithm, which is a backtracking algorithm, is always more efficient in the case where the incidence structure is resolvable. It is believed however that the new algorithm, which rests on a clique finding algorithm, performs significantly better when the incidence structure is *not* resolvable.

New features:

– New function `AllResolutions` which finds all the resolutions in an incidence structure.