Creation of a Knowledge Framework for Non-Commutative Computer Algebra

Svetlana COJOCARU[1], Alexandru COLESNICOV[1], Albert Heinle[2], Viktor LEVANDOVSKYY[2], Ludmila MALAHOV[1], Grischa Studzinski[2], Victor UFNAROVSKI[3]

[1]*IMI AŞ RM,* [2]*RWTH Aachen,* [3]*Lunds tekniska högskola*

[1]*{Svetlana.Cojocaru,kae,mal}@math.md,*

[2]*{Albert.Heinle,Viktor.Levandovskyy,Grischa.Studzinski}@math.rwth-aachen.de,*

[3]*Victor.Ufnarovski@math.lth.se*

Abstract — **Knowledge framework for non-commutative computer algebra consists of systematically discovered, collected, and thoroughly studied examples kept in a central repository (database). These examples are used at benchmarking the computer algebra systems in computing the non-commutative Gröbner basis. Till now such framework existed only for commutative case.**

Index Terms — **Gröbner basis, knowledge framework, non-commutative computer algebra.**

I. INTRODUCTION

The creation of a knowledge framework aims at systematic discovery and study of examples, which leads us to the central repository of digital data. With the tools in the framework we provide automated testing and comparisons of the abilities of computer algebra systems.

At present, despite the constantly growing need, there is no canonical set of examples, used for benchmarking the non-commutative Gröbner bases.

For ideals in commutative algebras, there are several collections of examples, but, to the best of our knowledge, there exists only one framework, implemented in the SYMBOLICDATA project [17].

The necessity of such collection is obvious. For example, it took more than 10 years to apply the theory of general non-commutative Gröbner bases to differential and difference equations, although such possibility was evident to the developers of this theory. Meanwhile the absence of freely available certified and truly organized information and the set of usage examples resulted in the simultaneous isolated efforts of software development instead of the use and reuse of the functionality, already developed in the general non-commutative world.

One of successful projects, demonstrating the systematic approach to the whole set of problems, is the OREMODULES package[1] for MAPLE, which consists of implemented algorithms and a nice collection of examples and solved problems in the realm of mathematical physics and systems and control theory.

II. GRÖBNER BASIS AND ITS IMPORTANCE

Suppose we have a commutative or non-commutative polynomial ring over a field of coefficients. A lot of mathematical problems can be reduced to study of ideals in such rings. Correspondingly, study of these ideals has a lot of applications in many areas.

An ideal can be represented by a set of its generators. This presentation is not unique and can not be used to solve many natural problems. For example, how to detect if two given set of generators represent the same ideal?

Starting from a set of ideal generators, the process known as Buchberger’s algorithm can extend it to the Gröbner basis of this ideal [19]. (Further variations of the Buchberger’s algorithm and other algorithms are also known.) Then the reduced Gröbner basis can be found by deleting the polynomials that are combinations of other elements of the basis.

The reduced Gröbner basis with its elements divided by corresponding leading coefficients is unique for an ideal. This solves the problem of equality of ideals as well as many others.

Gröbner bases are applicable in many important mathematical and physical problems that can be expressed through systems of polynomial equations.

For example, the following very simple criteria are known: the system of polynomial equations is inconsistent if and only if its Gröbner basis contains a non-zero constant; such a system has finite number of solutions if and only if its Gröbner basis contains polynomials whose leading monomials are powers of only one variable, for each variable.

In mathematics, many problems in differential equations and finite differences, in different domains including but not restricted by graph coloring, integer programming, theorem proving, and cryptography can be solved using Gröbner basis. In applications, such domains as weather forecast or petroleum production (and many others) can be exampled. In physics, the non-commutative Gröbner basis is used as well as commutative in many problems and many domains, for example, in quantum and nuclear physics.

Several applications of non-commutative Gröbner bases are, among others:

- rewriting systems, algebra and ring theory [18], especially group algebras of finitely presented groups (Heyworth [13], Madlener and Reinert [16], etc.);
- representation theory of algebras [8, 9];
- cryptanalysis and cryptography, as described by, e.g.,

[1] http://www.math.rwth-aachen.de/OreModules

M. Kreuzer [1], S. Bulygin with T. Rai ("polly cracker" algorithms [5]), and others;

- H^∞ control theory, see, e.g., W. Hellton et al. [11];
- identities between special functions (P. Paule, F. Chyzak [6] and others);
- automatic theorem proving (B. Buchberger et al. [4]).

The academic bibliography on Gröbner basis can be found in [20].

III. THE STRATEGY OF ELABORATION

The knowledge framework development is planned in the following steps and directions:

- study of present abilities of SYMBOLICDATA and comparison with the actual and future needs of an abstract framework;
- covering the area of free algebras by extending of SYMBOLICDATA to the two-sided ideals in free non-commutative algebras; there are here two separate parts, devoted to homogeneous and inhomogeneous examples;
- filling the framework with examples (problems and canonical results for them);
- adopt the experience to the finitely presented and graded algebras.

The abstract knowledge framework consists of the following components:

- database of examples;
- conversion routines between the database format, based on XML, and corresponding formats of computer algebra systems;
- various tools for studying different aspects of computations (memory usage, criteria and strategies applied, reduced bases, etc.)
- intuitive command line and Web interfaces to the database and its tools.

We started from the current version of SYMBOLICDATA project and began to extend it with a non-commutative subproject. SYMBOLICDATA does not provide us with the functionality we need, so we need to develop the missing tools for the framework.

IV. COMPUTER ALGEBRA SYSTEMS

Below we enlist computer algebra systems, which provide a user with a possibility to perform computations in free associative algebras and path algebras. The following list of such systems, to the best of our knowledge, is exhaustive.

BERGMAN, by J. Backelin et al. [14], is a powerful and flexible tool to calculate Gröbner bases, Hilbert and Poincaré-Betti series, Anick resolutions, and Betti numbers in non-commutative algebras and in modules over them. Per default BERGMAN takes homogeneous polynomials as the input. However, one is able to compute Gröbner bases of non-homogeneous ideals using homogenization or so-called *rabbit strategy* provided by BERGMAN.

MAGMA, by J. Cannon, W. Bosma et al. [3] is, among other, a generalization of Buchberger's algorithm to one- and two-sided ideals of finitely presented *K*-algebras as well as a non-commutative generalization (due to Allan Steel) of the Faugère F4 algorithm. These developments are quite recent in MAGMA. There are basic ideal operations and very important vector enumeration tools implemented.

GBNP (also called GROBNER), by A. Cohen and D. Gijsbers [7], is a package for GAP 4 with the implementation of non-commutative Gröbner bases for free and path algebras, following the algorithmic approach of Mora. It is a recent development, gaining more and more functionality with every new release.

SINGULAR:LETTERPLACE is the very recent development by V. Levandovskyy and H. Schönemann, realized as kernel extension of computer algebra system SINGULAR. At present, SINGULAR:LETTERPLACE computes only with homogeneous input, but it uses a very different algorithm due to La Scala and Levandovskyy [15], which shows very impressive performance on the variety of hard examples.

FELIX, by J. Apel and U. Klaus [2], provides generalizations of Buchberger's algorithm to free *K*-algebras, polynomial rings and *G*-algebras. Also, syzygies computations and basic ideal operations are implemented.

NCGB, by J. W. Helton et al. [12], is a package for MATHEMATICA, partially written in C. It is a part of the NCALGEBRA suite, which performs various operations (e.g. simplification and reduction modulo the Gröbner basis) of non-commutative expressions.

OPAL, by B. Keller et al. [10], is the specialized standalone system for Gröbner bases in free and path algebras. OPAL does not require the homogeneity of an input and is able to compute degree-bounded Gröbner basis.

V. INFORMATIONAL SYSTEM ON NON-COMMUTATIVE COMPUTER ALGEBRA

The informational system we provide contains the data base of known (stable) non-commutative computer algebra systems with the complete description of their possibilities and restrictions.

Web sites that collect information on Computer Algebra Systems (CASs) exist. Several examples are:

- http://orms.mfo.de/ – Oberwolfach References on Mathematical Software by Prof. Dr. Gert-Martin Greuel et al., the Mathematical Research Institute Oberwolfach;
- http://www.risc.jku.at/Groebner-Bases-Implementations – Gröbner Bases Implementations, Functionality Check and Comparison by Viktor Levandovsky et al.);
- http://www.fachgruppe-computeralgebra.de/ – a site in German that contains a page with a (short) list of CASs).

We can also refer articles on CAS in general and on different CASs in Wikipedia.

In our database we provide rather complete information on specific CASs. The following data will be included for each CAS in the base: general information; downloading and installation information; main applications; commutativity aspects; specific calculations; programming language aspects; interface aspects; representation of main data structures; representation of main objects; operations; main procedures; orderings; processing aspects.

All data are described in details. For example, the general information includes: name, acronyms; web-address and mirrors; list of developers; history (date of first version, previous names); mode of distribution (free or commercial); the language in which the system was written; is the system standalone or it is a package for other system; is the system dependent of other commercial products (e.g., MAPLE); existence of a demo-version; current state (e.g., the system has a stable version and its development continues; the system has a stable supported version but is no more under development; the system is stable but not supported; only older versions exist, not working on the modern platforms, etc.).

We shall mention that almost all fields contain two levels of answers:

- primary (e.g, yes/no/maybe/do not know);
- detailed, which gives more details in case the answer is yes or maybe.

The system is available at http://www.math.md/nccas/.

VI. PRESENTATION OF EXAMPLES DATA

The usual problem can be formulated as follows. Given a computable field K and a free associative algebra $A = K<x_1, \ldots, x_n>$, one is interested in computing a two-sided Gröbner basis (possibly up to a fixed degree d) of an ideal $I = <f_1, \ldots, f_m>$ for $f_i \in A$ with respect to a fixed monomial ordering on A. A computable field has a prime subfield k, which is either $\mathbf{Q}$ or $\mathbf{Z}_\mathrm{p}$ for a prime p. K is then either a simple algebraic extension of k, defined by a parameter q subject to a minimal polynomial or a transcendental extension of k by multiple parameters $q_1,\ldots,q_L$.

The input information can be presented in finite terms. For two-sided ideal, we need to provide the description of the underlying free algebra, the degree bound (0 if no bound is set), and a set of polynomials being the ideal generators $f_1, \ldots, f_m$. The description of the free algebra consists of the description of underlying coefficient field, the set of polynomial variables $x_1, \ldots, x_n$, and the ordering of monomials: left (right) (weighted-)degree (reverse-) lexicographic, or similar. Finally, the coefficient field is described by the prime p, or parameters q or $q_1,\ldots,q_L$, with the minimal polynomial (for q only): $q^s + c_1 q^{(s-1)} + ... + c_s$.

All the polynomials are written in a notation, similar to LATEX.

The output is just one entry containing the list of polynomials forming the two-sided Gröbner basis: $g_1(x)$, ..., $g_T(x)$. The output polynomials are sorted by a monomial ordering starting with the lowest terms. For total weighted degree orderings, the output polynomials will be automatically sorted by degree.

For one-sided ideal, we need to extend the record slightly. Based on the record above, we can formulate the left Gröbner basis computation for a left ideal from a finitely presented algebra as follows. An important assumption is that the left Gröbner basis is computed with respect to the same ordering as the two-sided Gröbner basis of the two-sided ideal of relations. Moreover, the latter Gröbner basis must be finite. Thus, a finitely presented algebra is described by a free algebra $F = K<x_1,\ldots,x_n>$ over a field K, an ordering $<$ on F and a finite set of polynomials, which constitute a two-sided Gröbner basis of *relations*. Note, that in this case no degree truncation is allowed, that is in such a situation the degree bound must be 0.

In addition to the two-sided description, we must provide the generators of the left ideal as a set of polynomials.

We suppose that Gröbner bases of both two-sided ideal of relations and left ideal over the corresponding factor algebra are finite.

Then it is known, that by computing a completely reduced Gröbner basis for a fixed ordering and normalizing its generators by dividing out leading coefficients, we get the unique Gröbner basis for a fixed ordering. Hence it makes sense to supply each problem with the fixed ordering with the answer, which is unique by the arguments before. This makes the check of correctness of computation an effective procedure.

The validation check becomes much harder (or even impossible), if an ideal is given via inhomogeneous relations and no monomial ordering leads to a finite Gröbner basis. Introducing a degree bound in this case does not help much because of the absence of the graded structure. Hence, at the time being one can process only finite Gröbner basis for a general example. However, we are investigating various notions of truncated Gröbner basis for possibly overcoming this serious difficulty.

It seems reasonable to store both input and output data in XML-like format.

VII. EVALUATION PROCEDURE

A usual “example evaluation” procedure can be sketched as follows

1. Selection
   - select an example *ExampleX* by name, specify the options
   - select a computer algebra system *System1*
   - create an input file *Input-X-1* for *System1* from the *ExampleX*
   - run *System1* on *Input-X-1* and obtain
     - the output *Output-X-1*
     - total running time *Time-X-1*
     - in the case of error, error code *Error-X-1* and its description *ErrorDesc-X-1*
     - other auxiliary information
2. Validation (check of correctness)
   - extract the result from *Output-X-1*
   - convert the result in the database format
   - compare it with the precomputed result *ResultExampleX* by using earlier defined system *DefaultSystem*
   - report success; otherwise, output differences.
3. On successful validation: output total running time *Time-X-1*

VIII. FURTHER DEVELOPMENTS

The testing subjects can be expanded from Gröbner bases for two-sided ideals to the following:

- treating submodules of free modules including ideals;
- left (right) Gröbner bases over factor algebras modulo two-sided ideals (see above);
- graded Hilbert function and Poincare series for homogeneous input;
- *K*-dimension of finite dimensional objects (both one- and two-sided);
- syzygy modules, etc.

ACKNOWLEDGEMENTS: We are grateful to Hans-Gert Gräbe for maintaining and developing SYMBOLICDATA project over years and for numerous discussions on the topics. We also thank Benjamin Schnitzler (RWTH Aachen) for his help in implementation of the FreeAlgebra component.

REFERENCES

[1] P. Ackermann and M. Kreuzer. Gröbner basis cryptosystems. Appl. Algebra Eng. Commun. Comput. 17(3–4):173–194, 2006.

[2] Apel, J. and Klaus, U. FELIX, a Special Computer Algebra System for the Computation in Commutative and Non-commutative Rings and Modules, 1998. Available from: http://felix.hgb-leipzig.de/.

[3] W. Bosma, J. Cannon, and C. Playoust. The MAGMA algebra system. I: The user language. J. of Symbolic Computation, 24(3–4):235–265, 1997.

[4] B. Buchberger, A. Crăciun, T. Jebelean, L. Kovács, T. Kutsia, K. Nakagawa, F. Piroi, N. Popov, J. Robu, M. Rosenkranz, and W. Windsteiger. Theorema: Towards computer-aided mathematical theory exploration. J. Appl. Log., 4(4):470–504, 2006.

[5] S. Bulygin and T. Rai. Noncommutative Polly Cracker-type cryptosystems and chosen-ciphertext security, 2008.

[6] F. Chyzak and B. Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. J. of Symbolic Computation, 26(2):187–227, 1998.

[7] Cohen, A.M. and Gijsbers D.A.H. GBNP, a Non-commutative Gröbner Bases Package for GAP 4, 2003. Available from http://www.win.tue.nl/~amc/pub/grobner/.

[8] D. J. Green. Gröbner bases and the computation of group cohomology. Lecture Notes in Mathematics 1828. Springer, 2003.

[9] Green, E. Noncommutative Groebner bases, and projective resolutions. In: Draexler, P., ed., Computational methods for representations of groups and algebras. Proc. of the Euroconference in Essen, Germany, April, pages 29–60. Birkhaeuser, 1999.

[10] Green, E., Heath, L., and Keller, B. OPAL: A System for Computing Noncommutative Gröbner Bases. In: RTA '97: Proceedings of the 8th International Conference on Rewriting Techniques and Applications, pages 331–334. Springer, 1997.

[11] J. Helton and O. Merino. Classical control using H^∞ methods. Theory, optimization, and design. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. xvi, 292 p., 1998.

[12] Helton, J. W. and Stankus, M. NCGB 3.1, a Noncommutative Gröbner Basis Package for MATHEMATICA, 2001. Available from:
http://www.math.ucsd.edu/~ncalg/.

[13] A. Heyworth. Rewriting as a special case of non-commutative Gröbner basis theory. In: M.E.A. Atkinson, ed., Computational and geometric aspects of modern algebra, pp. 101–105. Cambridge University Press, 2000.

[14] J. Backelin et al. The Gröbner basis calculator BERGMAN, 2006. Available from:
http://servus.math.su.se/bergman/.

[15] R. La Scala and V. Levandovskyy. Letterplace ideals and non-commutative Gröbner bases. J. of Symbolic Computation, Volume 44, Issue 10, October, 2009, pp. 1374–1393.

[16] B. Reinert. On Gröbner bases in Monoid and Group Rings. Doctoral Thesis, Universität Kaiserslautern, 1995.

[17] The SYMBOLICDATA Project, 2000–2011. Available from: http://www.SymbolicData.org.

[18] V. Ufnarovski. Combinatorial and Asymptotic Methods of Algebra. In: volume 57 of Algebra-VI (A.I. Kostrikin and I.R. Shafarevich, Eds), Encyclopedia of Mathematical Sciences. Springer, 1995.

[19] D. Cox, J. Little, and D. O'Shea: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, New York, 1992.

[20] http://www.ricam.oeaw.ac.at/Groebner-Bases-Bibliography/