General Polynomial Reduction with TH∃OREM∀ Functors: Applications to Integro-Differential Operators and Polynomials

Bruno Buchberger Georg Regensburger Markus Rosenkranz Loredana Tec

General Polynomial Reduction. We outline a prototype implementation of the *algorithms for integro-differential operators/polynomials* in [12]. Our approach based on a generic implementation of noncommutative monoid rings with reduction, programmed in the functors language of the TH∃OREM∀ system. The integro-differential operators—realized by a suitable quotient of noncommutative polynomials over a given integro-differential algebra—can be used for solving and manipulating boundary problems for linear ordinary differential equations. For describing extensions of integro-differential algebras algorithmically, we use integro-differential polynomials.

We use a fixed *Gröbner basis* for normalizing integro-differential operators. Gröbner bases were invented by Buchberger [2, 3] for commutative polynomials and reinvented in [1] for noncommutative ones. While [9] analyzes the computational aspects of the latter, it does not support two features that are important for our present setting: the usage of infinitely many indeterminates and reduction modulo an (algorithmic) infinite system of polynomials.

Among the *systems implementing noncommutative Gröbner bases*, most address certain special classes (e.g. algebras of solvable type or homogeneous polynomials) which do not include our present case. To our best knowledge, none of these allow polynomials with infinitely many indeterminates and reduction modulo an infinite system of polynomials. For details, see the website `http://www.ricam.oeaw.ac.at/Groebner-Bases-Implementations`.

Our implementation, although targeted at the integro-differential applications described below, follows a generic approach that encompasses commutative/noncommutative polynomials as well as one/two-sided reduction. Polynomial rings are formulated as *monoid rings* (leading to the standard commutative or noncommutative polynomials by employing the additive monoid $\mathbb{N}^n$ or the word monoid $\{x_1, \ldots, x_n\}^*$, respectively), while polynomial reduction is realized by a noncommutative adaption of *reduction rings* (rings with so-called reduction multipliers) in the sense of [4, 14]; for a noncommutative approach along different lines, we refer to [8].

The TH∃OREM∀ Functor Language. The generic implementation of monoid rings with reduction multipliers is realized through *functors* whose principle and implementation in the TH∃OREM∀ version of higher order predicate logic were introduced by B. Buchberger. The general idea—and its use for structuring those domains in which Groebner bases computation is possible—is described in [4, 5], where you also find references to original and early papers by B. Buchberger on the subject. For a general discussion of functor programming, see also [15].

The TH∃OREM∀ system is designed as an integrated environment for doing mathematics [6], in particular proving, computing, and solving in various domains of mathematics. Its core language is higher-order predicate logic, containing a natural *programming language* such that algorithms can be coded and verified in a unified formal frame. Functors are a powerful tool for realizing a modular and generic build-up of hierarchical domains in mathematics. For speeding up computations, one may also use the new Java-to-Theorema compiler described in the recent thesis [16].

Starting from the base category of rings and monoids, the *monoid ring* is the crucial functor that builds up polynomials. After adding reduction multipliers, the functions for enumerating Groebner bases are added by virtue of an extension functor (a functor that leaves previous operations unchanged and adds new ones). Obviously, the coefficient rings may in turn be monoid rings, and there are also various functors for composing useful monoid rings: word monoids, cartesian products, free products.

Anticipating the explanations given in the following two sections, let us here state one *interesting example* of a chain of functors that can be realized in our system: integro-differential operators over exponential polynomials with an undetermined function. This proceeds in three stages: Starting with say $\mathbb{Q}$, the exponential polynomials are obtained as the monoid ring with $\mathbb{N} \times \mathbb{Q}$ as monoid. The second step is the functor of integro-differential polynomials, the last step the functor of integro-differential operators.

Integro-Differential Operators. The notion of integro-differential operators [13] is a generalization of the "Green's polynomials" of [11]. They can be seen as an algebraic analog of differential, integral and boundary operators in the context of linear ordinary differential equations (LODEs). They are particularly useful for treating

boundary problems for LODEs as they express both the problems statement (differential equation and boundary conditions) and its solution operator (an integral operator usually called "Green's operator"). We have described methods for solving and factoring boundary problems in [13] and an abstract (inconstructive) framework in [10].

The *implementation* of integro-differential operators proceeds along the lines sketched above: A monoid ring is built up for the word monoid over the infinite alphabet consisting of the letters ∂ and $\int$ along with all basis elements $x^n e^{\lambda x}$ ($n \in \mathbb{N}, \lambda \in \mathbb{Q}$) of the exponential polynomials. Then we factor out the nine (parametrized) rewrite rules described in Table 1 of [13, 12], which form a Gröbner basis in the underlying polynomial ring. This is achieved by working with the normal forms of the polynomial reduction induced by these rules.

Integro-Differential Polynomials. The integro-differential polynomials over a given integro-differential algebra, introduced in [12], have a slightly different flavor: They form a *commutative algebra* (since multiplication is understood pointwise and not by composition as above) modelling nonlinear differential and integral operators with an indeterminate u, so a typical element would be $\int(x^4 u u'' {}^2 \int(x e^{3x} u^2 u' {}^3 \int u))$. Another interpretation is that they are certain functions (of x) involving an "unknown" function (namely u). This means one can describe extensions of a given integro-differential algebra by forming suitable quotients of the integro-differential polynomials over it.

An elegant abstract characterization of integro-differential polynomials is in terms of a general polynomial algebra [7] in the variety of integro-differential algebras. For *computational purposes*, we have introduced a canonical simplifier for the induced congruence (identifying different expressions denoting the same integro-differential polynomial), which thus solves the word problem in this variety. The resulting functor is somewhat similar to the monoid ring (where polynomial reduction provides a canonical simplifier), but some operations need special attention: the multiplication of canonical forms involves the shuffle product, and the integral must be computed by a careful case distinction on the differential exponents.

References

[1] G. M. Bergman. The diamond lemma for ring theory. *Adv. in Math.*, 29(2):179–218, 1978.

[2] B. Buchberger. *An Algorithm for Finding the Bases Elements of the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal (German).* PhD thesis, Univ. of Innsbruck, 1965. English translation published in *J. Symbolic Comput.*, 41(3-4):475–511, 2006.

[3] B. Buchberger. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. *Aequationes Math.*, 4:374–383, 1970. English translation in B. Buchberger, F. Winkler (eds.), Gröbner Bases and Applications, Cambridge University Press, 1998.

[4] B. Buchberger. Groebner rings and modules. In S. Maruster, B. Buchberger, V. Negru, and T. Jebelean, editors, *Proceedings of SYNASC 2001*, pages 22–25, 2001.

[5] B. Buchberger. Groebner bases in theorema using functors. In J. Faugere and D. Wang, editors, *Proceedings of SCC'08*, pages 1–15. LMIB Beihang University Press, 2008.

[6] B. Buchberger et al. Theorema: Towards computer-aided mathematical theory exploration. *J. Appl. Log.*, 4(4):359–652, 2006.

[7] H. Lausch and W. Nöbauer. *Algebra of polynomials.* North-Holland Publishing Co., Amsterdam, 1973. North-Holland Mathematical Library, Vol. 5.

[8] K. Madlener and B. Reinert. Non-commutative reduction rings. *Rev. Colombiana Mat.*, 33(1):27–49, 1999.

[9] T. Mora. An introduction to commutative and noncommutative Gröbner bases. *Theoret. Comput. Sci.*, 134(1):131–173, 1994.

[10] G. Regensburger and M. Rosenkranz. An algebraic foundation for factoring linear boundary problems. *Ann. Mat. Pura Appl. (4)*, 2008. DOI:10.1007/s10231-008-0068-3.

[11] M. Rosenkranz. A new symbolic method for solving linear two-point boundary value problems on the level of operators. *J. Symbolic Comput.*, 39(2):171–199, 2005.

[12] M. Rosenkranz and G. Regensburger. Integro-differential polynomials and operators. In D. Jeffrey, editor, *Proceedings of ISSAC'08*. ACM Press, 2008.

[13] M. Rosenkranz and G. Regensburger. Solving and factoring boundary problems for linear ordinary differential equations in differential algebras. *J. Symbolic Comput.*, 43(8):515–544, 2008.

[14] S. Stifter. Gröbner bases of modules over reduction rings. *J. Algebra*, 159(1):54–63, 1993.

[15] W. Windsteiger. Building up hierarchical mathematical domains using functors in THEOREMA. In A. Armando and T. Jebelean, editors, *ENTCS*, volume 23, pages 401–419. Elsevier, 1999.

[16] A. Zapletal. *Compilation of Theorema Programs*. PhD thesis, Research Institute for Symbolic Computation (RISC), Johannes Kepler University, Linz, Austria, June 2008.