Factoring Elements in G-Algebras with `ncfactor.lib`

ICMS 2016 – Berlin – Germany

Albert Heinle

Symbolic Computation Group
David R. Cheriton School of Computer Science
University of Waterloo
Canada

2016–07–12

Introduction

G-Algebras

Definition

For $n \in \mathbb{N}$ and $1 \leq i < j \leq n$ consider the units $c_{ij} \in \mathbb{K}^*$ and polynomials $d_{ij} \in \mathbb{K}[x_1, \ldots, x_n]$. Suppose, that there exists a monomial total well-ordering $\prec$ on $\mathbb{K}[x_1, \ldots, x_n]$, such that for any $1 \leq i < j \leq n$ either $d_{ij} = 0$ or the leading monomial of d_{ij} is smaller than $x_i x_j$ with respect to $\prec$. The $\mathbb{K}$-algebra $A := \mathbb{K}\langle x_1, \ldots, x_n \mid \{x_j x_i = c_{ij} x_i x_j + d_{ij} : 1 \leq i < j \leq n\}\rangle$ is called a G-**algebra**, if $\{x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n} : \alpha_i \in \mathbb{N}_0\}$ is a $\mathbb{K}$-basis of A.

Remark

- *Also known as "algebras of solvable type" and "PBW (Poincaré Birkhoff Witt) Algebras"*

Definition

If $c_{ij} = 1$ for all i, j in the definition above, then we call the resulting $\mathbb{K}$ algebra a G-**algebra of Lie type**.

Examples for G-Algebras

- Weyl algebras ($\mathbb{K}\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \mid \forall i : \partial_i x_i = x_i \partial_i + 1\rangle$)
- Shift algebras ($\mathbb{K}\langle x_1, \ldots, x_n, s_1, \ldots, s_n \mid \forall i : s_i x_i = (x_i + 1) s_i\rangle$)
- q-Weyl algebras
  ($\mathbb{K}\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \mid \forall i \exists q_i \in \mathbb{K}^* : \partial_i x_i = q_i x_i \partial_i + 1\rangle$)
- q-Shift algebras
  ($\mathbb{K}\langle x_1, \ldots, x_n, s_1, \ldots, s_n \mid \forall i \exists q_i \in \mathbb{K}^* : s_i x_i = q_i x_i s_i\rangle$)
- Universal enveloping algebras of finite dimensional Lie algebras.
- ...

Available Software for G-Algebras

- SAGE (package `ore_algebra`, Kauers et al. (2014)): Any G-algebra (and more) can be defined. (depending on SAGE version; no factorization algorithm provided)
- SINGULAR:PLURAL (Greuel et al. (2010)): Any G-algebra can be defined (factorization functionality via `ncfactor.lib`).
- REDUCE (package `NCPOLY`, Melenk and Apel (1994)): Supports G-algebras of Lie type (factorization algorithm provided).
- MAPLE:
  - Package `OreTools` (Abramov et al. (2003)): Single Ore-extensions
  - Package `Ore_algebra`: Defining non-commutative rings using pairs of non-commuting variables.
  - Factorization algorithm only for Weyl algebras (via the package `DETools`, van Hoeij (1997)).

Available Software for G-Algebras

- SAGE (package `ore_algebra`, Kauers et al. (2014)): Any G-algebra (and more) can be defined. (depending on SAGE version; no factorization algorithm provided)
- SINGULAR:PLURAL (Greuel et al. (2010)): Any G-algebra can be defined (factorization functionality via `ncfactor.lib`).← That is us
- REDUCE (package `NCPOLY`, Melenk and Apel (1994)): Supports G-algebras of Lie type (factorization algorithm provided).
- MAPLE:
  - Package `OreTools` (Abramov et al. (2003)): Single Ore-extensions
  - Package `Ore_algebra`: Defining non-commutative rings using pairs of non-commuting variables.
  - Factorization algorithm only for Weyl algebras (via the package `DETools`, van Hoeij (1997)).

Development History of `ncfactor.lib`

- In the beginning: First Weyl algebra, first shift algebra. Main ideas:
  - Shift algebra can be embedded in Weyl algebra.
  - $\mathbb{Z}$ graded structure on Weyl algebra utilized (weight vector $[-1, 1]$ for x, ∂).
  - Factorization of homogeneous elements $\rightarrow$ factorization in $\mathbb{K}[\theta]$ (+minor combinatorics).
  - Factorization of general polynomials $\rightarrow$ by ansatz method (knowledge needed: only finitely many factorizations possible (Tsarev, 1996)).

Development History of `ncfactor.lib`

- In the beginning: First Weyl algebra, first shift algebra. Main ideas:
  - Shift algebra can be embedded in Weyl algebra.
  - $\mathbb{Z}$ graded structure on Weyl algebra utilized (weight vector $[-1, 1]$ for x, ∂).
  - Factorization of homogeneous elements $\rightarrow$ factorization in $\mathbb{K}[\theta]$ (+minor combinatorics).
  - Factorization of general polynomials $\rightarrow$ by ansatz method (knowledge needed: only finitely many factorizations possible (Tsarev, 1996)).
- Then: First q-Weyl algebra:
  - Same $\mathbb{Z}$-graded structure as for Weyl algebra.
  - Similar methods for homogeneous elements.
  - General polynomials $\rightarrow$???

Development History of `ncfactor.lib`

- In the beginning: First Weyl algebra, first shift algebra. Main ideas:
  - Shift algebra can be embedded in Weyl algebra.
  - $\mathbb{Z}$ graded structure on Weyl algebra utilized (weight vector $[-1, 1]$ for x, ∂).
  - Factorization of homogeneous elements $\rightarrow$ factorization in $\mathbb{K}[\theta]$ ($+$minor combinatorics).
  - Factorization of general polynomials $\rightarrow$ by ansatz method (knowledge needed: only finitely many factorizations possible (Tsarev, 1996)).
- Then: First q-Weyl algebra:
  - Same $\mathbb{Z}$-graded structure as for Weyl algebra.
  - Similar methods for homogeneous elements.
  - General polynomials $\rightarrow$???
- Then: n^{th} Weyl algebras and n^{th} shift algebras.
  - Extension to $\mathbb{Z}^n$ graded structure was possible.
  - Need for proof that n^{th} Weyl and shift algebras only have finitely many possible factorizations.

Development History of `ncfactor.lib`

- In the beginning: First Weyl algebra, first shift algebra. Main ideas:
  - Shift algebra can be embedded in Weyl algebra.
  - $\mathbb{Z}$ graded structure on Weyl algebra utilized (weight vector $[-1, 1]$ for x, ∂).
  - Factorization of homogeneous elements $\rightarrow$ factorization in $\mathbb{K}[\theta]$ ($+$minor combinatorics).
  - Factorization of general polynomials $\rightarrow$ by ansatz method (knowledge needed: only finitely many factorizations possible (Tsarev, 1996)).
- Then: First q-Weyl algebra:
  - Same $\mathbb{Z}$-graded structure as for Weyl algebra.
  - Similar methods for homogeneous elements.
  - General polynomials $\rightarrow$???
- Then: n^{th} Weyl algebras and n^{th} shift algebras.
  - Extension to $\mathbb{Z}^n$ graded structure was possible.
  - Need for proof that n^{th} Weyl and shift algebras only have finitely many possible factorizations.

Finite Factorization Domain

Definition (Non-Commutative FFD, cf. (Bell et al., 2014))

Let A be a (not necessarily commutative) domain. We say that A is a **finite factorization domain** (FFD, for short), if every nonzero, non-unit element of A has at least one factorization into irreducible elements and there are at most finitely many distinct factorizations into irreducible elements up to multiplication of the irreducible factors by central units in A.

Remark

Classically, different factorizations in non-commutative rings are studied with respect to **similarity***: For a ring R, two elements $a, b \in R$ are said to be* **similar***, if R/aR and R/bR are isomorphic as left R-modules.*

Finite Factorization Domain

Definition (Non-Commutative FFD, cf. (Bell et al., 2014))

Let A be a (not necessarily commutative) domain. We say that A is a **finite factorization domain** (FFD, for short), if every nonzero, non-unit element of A has at least one factorization into irreducible elements and there are at most finitely many distinct factorizations into irreducible elements up to multiplication of the irreducible factors by central units in A.

Remark

Classically, different factorizations in non-commutative rings are studied with respect to **similarity***: For a ring R, two elements $a, b \in R$ are said to be* **similar***, if R/aR and R/bR are isomorphic as left R-modules. However, it is a very weak property, as one can e.g. see in (Giesbrecht and Heinle, 2012).*

G-Algebras are FFD

Theorem (cf. (Bell et al., 2014))

Let $\mathbb{K}$ be a field. Then G-algebras over $\mathbb{K}$ and their subalgebras are finite factorization domains.

Consequences

- We have now more than just the similarity property to characterize factorizations in G-algebras.
- New algorithmic problem: Calculate all factorizations of an element in a given G-algebra.
- With this knowledge, study how algorithms from commutative algebra can be generalized to certain non-commutative algebras.

The New Powers of `ncfactor.lib`

What `ncfactor.lib` can do...

- Factor elements in all G-algebras, with the following assumption on the underlying field $\mathbb{K}$:
  - Factorization must be implemented in SINGULAR for $\mathbb{K}[x_1, \ldots, x_n]$.
- Currently, this only excludes fields represented by floating point numbers and finite fields that are not prime (i.e. those of order p^k with p prime and $k > 1$).
- Practical examples of underlying fields where we can factor:
  - $\mathbb{Q}$, and any field extension of $\mathbb{Q}(\alpha)$ with some algebraic α.
  - $\mathbb{K}(x_1, \ldots, x_n)$ for $x_1, \ldots, x_n$ being transcendental, and $\mathbb{K}$ an already supported field.
- Calling the function `ncfactor` is enough. As a preprocessing, it will check if a better algorithm for this specific algebra is available and forward the input there.

What `ncfactor.lib` cannot do...

- Whatever non-commutative ring cannot be directly defined in SINGULAR:PLURAL:
  - Ore extensions of the form $\mathbb{K}[x;\sigma,\delta]$, where σ and δ map elements in $\mathbb{K}$ (Caruso and Borgne (2012) have a good implementation for that, with implementation of factorization algorithm by Giesbrecht (1998)).
  - Factorize elements in factor rings of G-algebras with respect to two-sided ideals.
  - Non-commutative rings with zero-divisors (like the integro-differential operators).
- G-algebras over a field $\mathbb{K}$, for which elements in $\mathbb{K}[x_1,\ldots,x_n]$ cannot be factored in SINGULAR:PLURAL.
- Factor elements in free algebras
- Generally scale to larger powers for arbitrary G-algebras.

What `ncfactor.lib` cannot do...

- Whatever non-commutative ring cannot be directly defined in SINGULAR:PLURAL:
  - Ore extensions of the form $\mathbb{K}[x; \sigma, \delta]$, where σ and δ map elements in $\mathbb{K}$ (Caruso and Borgne (2012) have a good implementation for that, with implementation of factorization algorithm by Giesbrecht (1998)).
  - Factorize elements in factor rings of G-algebras with respect to two-sided ideals.
  - Non-commutative rings with zero-divisors (like the integro-differential operators).
- G-algebras over a field $\mathbb{K}$, for which elements in $\mathbb{K}[x_1, \ldots, x_n]$ cannot be factored in SINGULAR:PLURAL.
- Factor elements in free algebras yet.
- Generally scale to larger powers for arbitrary G-algebras.

Functions Overview

- `facWeyl`: Returns all factorizations of elements in Weyl algebras using the algorithm described in (Giesbrecht et al., 2015).
- `facShift`: Returns all factorizations of elements in shift algebras via embedding in Weyl algebras.
- `facSubWeyl`: Returns all factorizations of elements in Weyl algebras which are embedded in a larger ring (comfort function).
- `homogFacNthQWeyl[_all]`: Returns one (resp. all) factorization of a $\mathbb{Z}^n$ homogeneous element in the n^{th} q-Weyl algebra.
- `ncfactor`: Returns all factorizations of elements in any supported G-algebra. Automatically chooses a more specified algorithm when available (like e.g. for Weyl algebras).
- For legacy reasons, we still have `facFirstWeyl`, `facFirstShift`, `homogFacFirstQWeyl[_all]`. They just call their bigger siblings, i.e. they can be ignored.

Software Demonstration

Some New Applications

Factorized Gröbner bases – Commutative

- The factorized Gröbner approach has been studied extensively for the commutative case (Czapor, 1989b,a; Davenport, 1987; Gräbe, 1995a,b).
- Application: Obtaining triangular sets.
- Possible extension: Allowing constraints on the solutions.
- Implementations: e.g. in SINGULAR and REDUCE.
- Idea: For each factor $\tilde{g}$ of a reducible element g during a Gröbner computation, recursively call algorithm on the same generator set, with g being replaced by $\tilde{g}$.

Generalization to Non-Commutative Rings

- Ideals in commutative ring $\leftrightarrow$ Varieties
- Ideals in Non-Commutative ring $\leftrightarrow$ Solutions
- Formal notion of solutions: Let $\mathcal{F}$ be a left A-module for a $\mathbb{K}$-algebra A (space of solutions). Let a left A-module M be finitely presented by an $n \times m$ matrix P. Then

$$\mathrm{Sol}_A(P, \mathcal{F}) = \{f \in \mathcal{F}^m : Pf = 0\}$$

- Divisors for commutative rings $\leftrightarrow$ Right divisors for non-commutative rings.

Picking the Right Right Divisors

There are different strategies:

- Split Gröbner computation with respect to different irreducible right divisors.

Picking the Right Right Divisors

There are different strategies:

- Split Gröbner computation with respect to different irreducible right divisors. $\Rightarrow$ This approach may cause lost of possible solutions to the whole system.

Picking the Right Right Divisors

There are different strategies:

- Split Gröbner computation with respect to different irreducible right divisors. $\Rightarrow$ This approach may cause lost of possible solutions to the whole system.
- Split Gröbner computation with respect to all possible maximal right divisors.

Picking the Right Right Divisors

There are different strategies:

- Split Gröbner computation with respect to different irreducible right divisors. $\Rightarrow$ This approach may cause lost of possible solutions to the whole system.
- Split Gröbner computation with respect to all possible maximal right divisors. $\Rightarrow$ Less possible solutions may be lost.

Picking the Right Right Divisors

There are different strategies:

- Split Gröbner computation with respect to different irreducible right divisors. $\Rightarrow$ This approach may cause lost of possible solutions to the whole system.
- Split Gröbner computation with respect to all possible maximal right divisors. $\Rightarrow$ Less possible solutions may be lost.
- Split Gröbner computation with respect to all possible non-unique maximal right divisors.

Picking the Right Right Divisors

There are different strategies:

- Split Gröbner computation with respect to different irreducible right divisors. $\Rightarrow$ This approach may cause lost of possible solutions to the whole system.
- Split Gröbner computation with respect to all possible maximal right divisors. $\Rightarrow$ Less possible solutions may be lost.
- Split Gröbner computation with respect to all possible non-unique maximal right divisors. $\Rightarrow$ Our choice!

Remark

This methodology also appears in the context of semifirs, where the concept of so called block factorizations or cleavages has been introduced to study the reducibility of a principal ideal (Cohn, 2006, Chapter 3.5).

Main Difference

In the commutative case, for an ideal I and the output $B_1, \ldots, B_m$ of the factorized Gröbner basis algorithm, one has

$$\sqrt{I} = \bigcap_{i=1}^{m} \sqrt{B_i}.$$

We would like to have something similar for the non-commutative case.

However, as the next example depicts, we do not have it.

Example I

Let

$$
\begin{aligned}
p = &(x^6 + 2x^4 - 3x^2)\partial^2 - (4x^5 - 4x^4 - 12x^2 - 12x)\partial \\
&+ (6x^4 - 12x^3 - 6x^2 - 24x - 12)
\end{aligned}
$$

in the polynomial first Weyl algebra. This polynomial appears in (Tsai, 2000, Example 5.7) and has two different factorizations, namely

$$
\begin{aligned}
p = &(x^4\partial - x^3\partial - 3x^3 + 3x^2\partial + 6x^2 - 3x\partial - 3x + 12)\cdot \\
&(x^2\partial + x\partial - 3x - 1) \\
= &(x^4\partial + x^3\partial - 4x^3 + 3x^2\partial - 3x^2 + 3x\partial - 6x - 3)\cdot \\
&(x^2\partial - x\partial - 2x + 4).
\end{aligned}
$$

Example II

A reduced Gröbner basis of
$\langle x^2\partial + x\partial - 3x - 1\rangle \cap \langle x^2\partial - x\partial - 2x + 4\rangle$, computed with
SINGULAR, is given by

$$\begin{aligned}\{&3x^5\partial^2 + 2x^4\partial^3 - x^4\partial^2 - 12x^4\partial + x^3\partial^2 - 2x^2\partial^3 + 16x^3\partial \\ &+ 9x^2\partial^2 + 18x^3 + 4x^2\partial + 4x\partial^2 - 42x^2 - 4x\partial - 12x - 12, \\ &2x^4\partial^4 - 2x^4\partial^3 + 11x^4\partial^2 + 12x^3\partial^3 - 2x^2\partial^4 - 2x^3\partial^2 \\ &+ 10x^2\partial^3 - 44x^3\partial - 17x^2\partial^2 + 64x^2\partial + 12x\partial^2 + 66x^2 \\ &+ 52x\partial + 4\partial^2 - 168x - 16\partial - 60\}.\end{aligned}$$

Remark

The space of holomorphic solutions of the differential equation associated to p in fact coincides with the union of the solution spaces of the two generators of the intersection.

Conclusion and Future Work

Future Work

- Latest `ncfactor.lib` can be found in the SINGULAR GitHub repository[1].
- More efficient algorithms and implementations to factor (certain) G-algebras.
- Categorization of rings with respect to the factorization properties of their elements (as e.g. done for commutative integral domains (Anderson et al., 1990; Anderson and Anderson, 1992; Anderson and Mullins, 1996; Anderson, 1997)).
- Study the output of non-commutative factorized Gröbner basis algorithm. What does it say about the ideal structure? What is the connection to the solution space?

[1] `https://github.com/Singular/Sources/blob/spielwiese/Singular/LIB/ncfactor.lib`

Bibliography I

Abramov, S. A., Le, H., and Li, Z. (2003). OreTools: A Computer Algebra Library for Univariate Ore Polynomial Rings. *School of Computer Science CS-2003-12, University of Waterloo*.

Anderson, D. (1997). *Factorization in integral domains*, volume 189. CRC Press.

Anderson, D. and Anderson, D. (1992). Elasticity of factorizations in integral domains. *Journal of pure and applied algebra*, 80(3):217–235.

Anderson, D., Anderson, D., and Zafrullah, M. (1990). Factorization in integral domains. *Journal of pure and applied algebra*, 69(1):1–19.

Anderson, D. and Mullins, B. (1996). Finite factorization domains. *Proceedings of the American Mathematical Society*, 124(2):389–396.

Bell, J. P., Heinle, A., and Levandovskyy, V. (2014). On noncommutative finite factorization domains. *To Appear in the Transactions of the American Mathematical Society; arXiv preprint arXiv:1410.6178*.

Caruso, X. and Borgne, J. L. (2012). Some Algorithms for Skew Polynomials over Finite Fields. *arXiv preprint arXiv:1212.3582*.

Cohn, P. M. (2006). *Free ideal rings and localization in general rings*, volume 3. Cambridge University Press.

Czapor, S. R. (1989a). Solving algebraic equations: combining Buchberger's algorithm with multivariate factorization. *Journal of Symbolic Computation*, 7(1):49–53.

Czapor, S. R. (1989b). Solving algebraic equations via Buchberger's algorithm. In *Eurocal'87*, pages 260–269. Springer.

Davenport, J. H. (1987). Looking at a set of equations. *Technical report, School of Mathematical Sciences, The University of Bath.*

Giesbrecht, M. (1998). Factoring in Skew-Polynomial Rings over Finite Fields. *Journal of Symbolic Computation*, 26(4):463–486.

Giesbrecht, M. and Heinle, A. (2012). A Polynomial-Time Algorithm for the Jacobson Form of a Matrix of Ore Polynomials. In *Computer Algebra in Scientific Computing*, pages 117–128. Springer.

Giesbrecht, M., Heinle, A., and Levandovskyy, V. (2015). Factoring linear partial differential operators in n variables. *Journal of Symbolic Computation*.

Bibliography II

Gräbe, H.-G. (1995a). On factorized Gröbner bases. In *Computer algebra in science and engineering, pages 77–89. World Scientific*. Citeseer.

Gräbe, H.-G. (1995b). *Triangular systems and factorized Gröbner bases*. Springer.

Greuel, G.-M., Levandovskyy, V., Motsak, A., and Schönemann, H. (2010). PLURAL. A SINGULAR 3.1 Subsystem for Computations with Non-commutative Polynomial Algebras. Centre for Computer Algebra, TU Kaiserslautern.

Kauers, M., Jaroschek, M., and Johansson, F. (2014). Ore Polynomials in Sage. In Gutierrez, J., Schicho, J., and Weimann, M., editors, *Computer Algebra and Polynomials*, Lecture Notes in Computer Science, pages 105–125.

Melenk, H. and Apel, J. (1994). Reduce package ncpoly: Computation in non-commutative polynomial ideals. *Konrad-Zuse-Zentrum Berlin (ZIB)*, 65.

Tsai, H. (2000). Weyl closure of a linear differential operator. *Journal of Symbolic Computation*, 29:747–775.

Tsarev, S. (1996). An Algorithm for Complete Enumeration of all Factorizations of a Linear Ordinary Differential Operator. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation 1996*. New York, NY: ACM Press.

van Hoeij, M. (1997). Factorization of Differential Operators with Rational Functions Coefficients. 24(5):537–561.