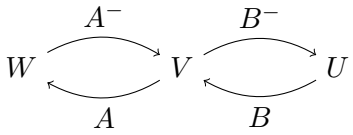
Certifying operator identities via noncommutative Gröbner bases*

Clemens Hofstadler, Clemens G. Raab, and Georg Regensburger
Johannes Kepler Universität Linz, 4040 Linz, Austria
clemens.hofstadler@liwest.at,{clemens.raab,georg.regensburger}@jku.at

Matrices or linear operators and their identities can be modelled algebraically by noncommutative polynomials in the free algebra. For proving new identities of matrices or operators from given ones, computations are done formally with noncommutative polynomials. Computations in the free algebra, however, are not necessarily compatible with formats of matrices resp. with domains and codomains of operators. For ensuring validity of such computations in terms of operators, in principle, one would have to inspect every step of the computation. In [9], an algebraic framework is developed that allows to rigorously justify such computations without restricting the computation to compatible expressions. The main result of that paper reduces the proof of an operator identity to verifying membership of the corresponding polynomial in the ideal generated by the polynomials corresponding to the assumptions and verifying compatibility of this polynomial and of the generators of the ideal.

Our MATHEMATICA package `OperatorGB` provides two main functionalities: certified ideal membership and automated compatibility checks. Certificates for ideal membership in the free algebra in the form of cofactor representations in terms of the generators of the ideal are computed via (partial) Gröbner bases. Compatibility of noncommutative polynomials is checked against a quiver specifying the domains and codomains of operators. The package is available at `http://gregensburger.com/softw/OperatorGB`

As a running example, we consider a simple statement about the product of two inner inverses. An operator A^- is called inner inverse (or g-inverse or $\{1\}$-inverse) of an operator A, if $AA^-A = A$. Assume, given linear operators $A : V \to W$ and $B : U \to V$ with inner inverses $A^- : W \to V$ and $B^- : V \to U$, we want to prove that B^-A^- is an inner inverse if A^-ABB^- is idempotent. The domains and codomains of the operators involved are encoded in the following diagram.



In terms of noncommutative polynomials in the indeterminates a, a^-, b, b^-, the assumptions and the claim of this statement correspond to

$$f_1 = aa^-a - a, \quad f_2 = bb^-b - b, \quad f_3 = a^-abb^-a^-abb^- - a^-abb^-, \quad \text{and} \quad f = abb^-a^-ab - ab.$$

1 Noncommutative Gröbner bases with cofactors

The free algebra $K\langle X\rangle$ on a set X over a field K can be viewed as the ring of noncommutative polynomials in the indeterminates X with coefficients in K, where indeterminates commute with coefficients but not with each other. The monomials are words $x_1 \dots x_n$ over the alphabet X, including the empty word 1. For a set $F \subseteq K\langle X\rangle$, we consider the (two-sided) ideal (F) in $K\langle X\rangle$, which is given by all polynomials of the form

*Supported by the Austrian Science Fund (FWF): P27229 and P31952

$f = \sum_i a_i f_i b_i$ where $f_i \in F$ and $a_i, b_i \in K\langle X\rangle$. We call such a representation of f a *cofactor representation* w.r.t. F, which serves as a certificate for ideal membership. Note that cofactors a_i and b_i from several summands with the same f_i in general cannot be collected on both sides of the f_i simultaneously.

For commutative polynomials over a field, ideal membership can be decided by Buchberger's algorithm [3] computing a Gröbner basis of the ideal. In contrast, ideal membership of noncommutative polynomials is undecidable in general. Still, a noncommutative analog of Buchberger's algorithm [8] can be used to enumerate a (potentially infinite) Gröbner basis. If the enumeration is stopped after finitely many steps without obtaining a Gröbner basis, then it is referred to as *partial Gröbner basis*. In practice, a partial Gröbner basis $G \subset (F)$ often suffices to verify ideal membership of a given polynomial $f \in K\langle X\rangle$ by reducing it to zero. If this is the case, then we obtain a cofactor representation of f w.r.t. G by keeping track of the cofactors in the reduction process. In order to obtain a certificate of ideal membership $f \in (F)$, which can be checked independently of the Gröbner basis computation, we need cofactor representations of the elements of G w.r.t. F in addition. This can be achieved by tracing the Gröbner basis computation.

Current implementations of computations with noncommutative polynomials and Gröbner bases of two-sided ideals generated by them are provided by the MATHEMATICA package `NCAlgebra` [5], by the `freegb.lib` library [7] of SINGULAR, by MAGMA [2], and by the `GBNP` package of GAP [4]. The latter also allows to compute cofactor representations, if the Gröbner basis is finite. Other software with support for Gröbner bases of noncommutative polynomials include `Bergman`, `Opal`, and `Felix`, see [6] for an overview.

In our package, we implement a noncommutative analog of Buchberger's algorithm. In each iteration of the main loop, all ambiguities of the current partial Gröbner basis are processed. The ambiguities defined by Bergman [1] correspond to what is called obstructions in [8] and automatically incorporate an analog of Buchberger's product criterion. Currently, we do not apply additional criteria for removing useless ambiguities. Note that, in contrast to the commutative case, two elements of the partial Gröbner basis can give rise to several ambiguities. For each ambiguity, the corresponding S-polynomial is generated and completely reduced by the current partial Gröbner basis and we keep track of the cofactors w.r.t. the current partial Gröbner basis. In our implementation, the number of iterations of the main loop can be limited directly or by imposing a degree bound on the ambiguities considered. After the main loop, cofactor representations of the elements of the final (partial) Gröbner basis G w.r.t. the original generators of the ideal F are computed based on the cofactors collected so far. Currently, the final output G is not autoreduced.

For reduction of a polynomial with respect to a given set of polynomials, we exploit MATHEMATICA's pattern matching capabilities. Each polynomial from the given set is transformed into a replacement rule, which replaces multiples of the leading term by corresponding multiples of the tail keeping track of cofactors used. The cofactors are stored in a dedicated list provided by the user. For representing and computing with noncommutative polynomials internally, we implemented a custom noncommutative product. Generating ambiguities and constructing S-polynomials automatically make use of multiple CPUs present. Parallelizing other parts of the computation did not result in a speedup for the examples we checked.

In the example above, we need to verify that f is an element of the ideal $(f_1, f_2, f_3) \subseteq \mathbb{Q}\langle a, a^-, b, b^-\rangle$. With our package, we first declare the ring by calling `SetUpRing[{a,a⁻,b,b⁻}]`, which by default uses the degree lexicographic monomial order. Then, we use the command `Groebner` to compute a Gröbner basis G, which in this case is finite and has 6 elements.

```
In[3]:= G = Groebner[cofactorsG, {a ** a⁻ ** a - a, b ** b⁻ ** b - b, a⁻ ** a ** b ** b⁻ ** a⁻ ** a ** b ** b⁻ - a⁻ ** a ** b ** b⁻}]

Out[3]= {-a + a ** a⁻ ** a, -b + b ** b⁻ ** b,
  -a⁻ ** a ** b ** b⁻ + a⁻ ** a ** b ** b⁻ ** a⁻ ** a ** b ** b⁻, -a ** b ** b⁻ + a ** b ** b⁻ ** a⁻ ** a ** b ** b⁻,
  -a⁻ ** a ** b + a⁻ ** a ** b ** b⁻ ** a⁻ ** a ** b, -a ** b + a ** b ** b⁻ ** a⁻ ** a ** b}
```

The command `ReducedForm` computes a reduced form r of f w.r.t. G and stores a cofactor representation of $f-r$ w.r.t. G in a list provided by the user. Then, by the command `Rewrite` this can be converted into a cofactor representation w.r.t. F, consisting of a list of triples $\{a_i, f_i, b_i\}$.

```
In[4]:= ReducedForm[cofactors1, G, a ** b ** b^- ** a^- ** a ** b - a ** b]
Out[4]= 0

In[5]:= Rewrite[cofactors1, cofactorsG]
Out[5]= {{a ** a^- ** a, -b + b ** b^- ** b, 1}, {-a ** a^- ** a ** b ** b^- ** a^- ** a, -b + b ** b^- ** b, 1},
         {a, -a^- ** a ** b ** b^- + a^- ** a ** b ** b^- ** a^- ** a ** b ** b^-, b},
         {-1, -a + a ** a^- ** a, b ** b^- ** a^- ** a ** b}, {1, -a + a ** a^- ** a, b}}
```

2 Compatibility with quivers

As mentioned in the beginning, in general, verifying ideal membership alone does not allow to conclude that the corresponding identity of operators can be proven from the assumptions in terms of operators. Next, we give a brief summary of the framework developed in [9], which allows to make this conclusion in general. For further details and proofs of more general statements, we refer to that paper.

A diagram describing domains and codomains of operators is formalized as a *labelled quiver* Q, i.e. a directed multigraph where edges have labels in X. Then, composition of operators corresponds to paths in Q. A polynomial in $K\langle X\rangle$ is called *compatible* with Q if all its monomials correspond to paths in Q with the same start and same end. We call a polynomial $f \in (F)$ a *Q-consequence* of F if it can be obtained from F by doing only computations with polynomials that are compatible with the labelled quiver Q. This means that the operator corresponding to f is obtained by a valid computation with operators. Since the operators corresponding to elements of F are zero, the operator corresponding to f is zero as well, i.e. the operator identity corresponding to f holds. The main result gives a simple characterization of those elements of the ideal (F) that are Q-consequences of F.

Theorem 1 *Let $F \subseteq K\langle X\rangle$ and $f \in (F)$. Then, for all labelled quivers Q such that all elements of F are compatible with Q we have that*

$$f \text{ is compatible with } Q \iff f \text{ is a } Q\text{-consequence of } F.$$

In our package, we use `Q={{a,V,W},{a^-,W,V},{b,U,V},{b^-,V,U}}` to specify the quiver for our example. Then, by the command `QSignature`, we obtain for each generator of the ideal the list of pairs on which the polynomial could be interpreted as an operator. In particular, a polynomial is compatible if and only if the list is not empty.

```
In[7]:= QSignature[{a ** a^- ** a - a, b ** b^- ** b - b, a^- ** a ** b ** b^- ** a^- ** a ** b ** b^- - a^- ** a ** b ** b^-}, Q]
Out[7]= {{{V, W}}, {{U, V}}, {{V, V}}}

In[8]:= QSignature[a ** b ** b^- ** a^- ** a ** b - a ** b, Q]
Out[8]= {{U, W}}
```

The command `Certify` provides a convenient way of computing the signatures of the polynomials F and f corresponding to the assumptions and the claim, a reduced form r of f modulo the the ideal (F), and a cofactor representation of the difference $f-r$ w.r.t. F, all in one go. If $r=0$, this certifies ideal membership $f \in (F)$, which together with the compatibility of F and f with the quiver gives a proof of the claim in terms of operators based on Theorem 1 above.

```
In[9]:= Certify[{a ** a^- ** a - a, b ** b^- ** b - b, a^- ** a ** b ** b^- ** a^- ** a ** b ** b^- - a^- ** a ** b ** b^-},
         a ** b ** b^- ** a^- ** a ** b - a ** b, Q]

Out[9]= {{{{V, W}}, {{U, V}}, {{V, V}}}, {{U, W}}, 0,
         {{a ** a^- ** a, -b + b ** b^- ** b, 1}, {-a ** a^- ** a ** b ** b^- ** a^- ** a, -b + b ** b^- ** b, 1},
          {a, -a^- ** a ** b ** b^- + a^- ** a ** b ** b^- ** a^- ** a ** b ** b^-, b},
          {-1, -a + a ** a^- ** a, b ** b^- ** a^- ** a ** b}, {1, -a + a ** a^- ** a, b}}}
```

3 Conclusion

For proving operator identities, the main goal is to find a cofactor representation of a given polynomial w.r.t. given generators of an ideal. So, a partial Gröbner basis is needed that allows to reduce the given polynomial to zero. Our heuristic choices for strategies of the computation (e.g. use of deletion criteria) have been guided by that. As a consequence, our implementation currently produces large partial Gröbner bases with few iterations by keeping "useless" pairs. These choices depend on the concrete cases at hand and will be subject of future investigation.

While the above proof of an operator identity is just a small illustrative example, our package can be used to certify more involved operator identities. In work in progress with our collaborators, the largest example we tried so far involved 30 binomial generators with maximal degree 20 in 18 indeterminates. With the degree bound 20 on the ambiguities, 6 iterations of the main loop took 81 seconds on a laptop with 4 CPUs. During this computation, more than 20,000 ambiguities were considered and the resulting partial Gröbner basis has 1087 elements. Their cofactor representations consist of up to 685 summands. The cost of computing cofactors is small as the computation without cofactors takes 61 seconds, which is comparable to the MATHEMATICA package `NCAlgebra`.

References

[1] George M. Bergman, *The diamond lemma for ring theory*, Adv. in Math. 29, pp. 178–218, 1978.

[2] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma Algebra System I: The User Language*, J. Symb. Comput. 24, pp. 235–265, 1997.

[3] Bruno Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, University of Innsbruck, 1965. English translation in J. Symbolic Comput. 134, pp. 475–511, 2006.

[4] The GAP Group, *GAP – Groups, Algorithms, and Programming*, Version 4.10.1, 2019.

[5] J. William Helton and Mark Stankus, *Computer assistance for "discovering" formulas in system engineering and operator theory*, J. Funct. Anal. 161, pp. 289–363, 1999.

[6] Roberto La Scala and Viktor Levandovskyy, *Letterplace ideals and non-commutative Gröbner bases*, J. Symb. Comput. 44, pp. 1374–1393, 2009.

[7] Viktor Levandovskyy, Grischa Studzinski, and Benjamin Schnitzler, *Enhanced Computations of Gröbner Bases in Free Algebras as a New Application of the Letterplace Paradigm*, Proc. ISSAC'13, pp. 259–266, 2013.

[8] Teo Mora, *An introduction to commutative and noncommutative Gröbner bases*, Theoret. Comput. Sci. 134, pp. 131–173, 1994.

[9] Clemens G. Raab, Georg Regensburger, and Jamal Hossein Poor. *Formal proofs of operator identities by a single formal computation.* In preparation.