

Article

SAGBI Bases in G -Algebras

Muhammad Abdul Basit Khan ¹, Junaid Alam Khan ¹ and Muhammad Ahsan Binyamin ^{2,*}

¹ Department of Mathematical Sciences, Institute of Business Administration, Karachi-75270, Pakistan; mabasit@iba.edu.pk (M.A.B.K.); jakhan@iba.edu.pk (J.A.K.)

² Department of Mathematics, GC University, Faisalabad-38000, Pakistan

* Correspondence: ahsanbinyamin@gmail.com

Received: 10 January 2019; Accepted: 8 February 2019; Published: 13 February 2019



Abstract: In this article, we develop the theory of SAGBI bases in G -algebras and create a criterion through which we can check if a set of polynomials in a G -algebra is a SAGBI basis or not. Moreover, we will construct an algorithm to compute SAGBI bases from a subset of polynomials contained in a subalgebra of a G -algebra.

Keywords: G -algebra; SAGBI normal form; SAGBI basis

1. Introduction

Our interest in the topic of this paper was inspired by the work of Levandovskyy [1]. In [1], the author developed the concept and computational criterion for computing Gröbner bases in G -algebra whenever these bases have Poincare-Birkhof-Witt (PBW) bases.

The popular PBW theorem is initially defined in [2] for a special Lie algebra, known as enveloping algebra over a condition of finite dimension. PBW theorem is one of the most important tools to study representation theory, theory of algebra, and rings. The notion, G -algebra developed by Apel [3] and Mora [4]. This algebra is quoted as algebra of solvable types [5–7] and PBW algebras [8]. These are also nice generalization of commutative algebras and widely used in non-commutative algebraic geometry [9].

Gordon proposed the idea of Gröbner bases in [10] in 1900 while Gröbner bases for commutative rings of polynomials over a field K were defined and developed by Buchberger [11] in 1965. The theory of Gröbner bases in a free associative algebra was developed by Kandri and Rody [5]. Gröbner bases in G -algebras over a field K were defined by Levandovskyy, he also developed a criterion for the existence of these bases and gave a method to compute them [1].

It is natural to make analogue of Gröbner bases of ideal in K -subalgebra; this work was done independently by Robbiano and Sweedler in [12] and Kapur and Madlener in [13]. These bases are known as SAGBI bases. In [14] Nordbeck developed the concept of SAGBI bases in free associative algebra and gave a method to compute them.

In this paper, we establish the theory of SAGBI bases in a general G -algebra over a field K ; and we also develop a computational criterion for its construction.

The sketch of this paper is as follows. In Section 2, we briefly describe the concept of a G -algebra and give some definitions that will be used, including the definition of a SAGBI basis in a G -algebra (Definition 4). In Section 3, we define the process of subalgebra reduction in a G -algebra and introduce the concept SAGBI normal form in a G -algebra. Also, we give an algorithm (Algorithm 1) to compute it and its consequences (Proposition 3.8). Finally, in Section 4, we give a SAGBI bases criterion, (Theorem 1) which determines whether a given set is a SAGBI basis. Based on this criterion, we give an algorithm (Algorithm 2) to compute them.

2. Definitions and Notations

In this section, first, we will introduce G-algebras and then we will review some basic terminologies related with it. G-algebras are significant in the study of non-commutative algebras. It has a wide application area. The theory of Gröbner bases are well developed for G-algebras. The corresponding algorithms are implemented in Singular [15]. Details can be found in [16].

Definition 1. $T_n = K\langle x_1, \dots, x_n \rangle$ be the free associative K-algebra, generated by $\{x_1, \dots, x_n\}$ over K. Let $c_{ij} \in K \setminus \{0\}$ and d_{ij} , denote the standard polynomials in T_n , where $1 \leq i < j \leq n$. Consider

$$A = K\langle x_1, \dots, x_n \mid x_{ji} = c_{ij} \cdot x_i x_j + d_{ij}, 1 \leq i < j \leq n \rangle$$

A is termed as a G-algebra, if these conditions hold:

1. There exists a monomial well-ordering $<$ on \mathbb{N}^n such that

$$\text{for all } i < j, \quad LM(d_{ij}) < x_i x_j$$

2. For all $1 \leq i < j < k \leq n$, the polynomial

$$c_{ik}c_{jk} \cdot d_{ij}x_k - x_kd_{ij} + c_{jk} \cdot x_jd_{ik} - c_{ij} \cdot d_{ik}x_j + d_{jk}x_i - c_{ij}c_{ik} \cdot x_i d_{jk}$$

reduces to 0 with respect to the relations of A.

A K-algebra A has a PBW basis, if $\{x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}\}$, known as a standard word-set, is a K-basis of A.

Proposition 1 ([1]). Let A be a G-algebra. Then it is an integral domain and has a PBW basis.

Let $A = K\langle x_1, \dots, x_n \mid x_j x_i = c_{ij} \cdot x_i x_j + d_{ij}, 1 \leq i < j \leq n \rangle$ be a G-algebra over the field K. As A has a PBW basis, we say standard monomial in A appears as $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ of this basis. The set $Mon(A)$ consist all standard monomials from A, that is,

$$Mon(A) = \{x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \mid \alpha_k \geq 0\}.$$

Now we introduce the notion of a monomial ordering in a G-algebra.

Definition 2. Let A be a G-algebra in n variables.

1. A total ordering $<$ on $Mon(A)$ is called a monomial ordering, if it is a well-ordering on $Mon(A)$ and for all $x^\alpha, x^\beta, x^\gamma \in Mon(A)$ if $x^\alpha < x^\beta$, then $x^{\alpha+\gamma} < x^{\beta+\gamma}$. If $<$ is a monomial ordering on $Mon(A)$ then $<$ is said to be a monomial ordering on A.
2. As we know, $Mon(A)$ forms a K-basis of A, therefore any non-zero element f in A could be uniquely written as $f = c_\alpha x^\alpha + g$ with $c_\alpha \in K \setminus \{0\}$ and x^α a monomial. Please note that for any non-zero term $c_\beta x^\beta$ of g, we have $x^\beta < x^\alpha$. The monomial $x^\alpha \in Mon(A)$ represents the leading monomial of f, denoted by $LM(f)$. Here $c_\alpha \in K \setminus \{0\}$ represents the leading coefficient of f, denoted by $LC(f)$.
3. Let $H \subset A$, the notation $K\langle H \rangle_A$ means the subalgebra S of A generated by H. It is the polynomials set in the H-variables in A.
4. For $H \subset A$, $m(H)$ denotes a monomial in terms of elements of H, we call it H-monomial. For $m(H) = h_{i1}h_{i2} \cdots h_{it}$, $h_{ij} \in H$ we define

$$\overline{LM}m(H) = LM(LM(h_{i1})LM(h_{i2}) \cdots LM(h_{it}))$$

Also,

$$\overline{LT}m(H) = LT(LT(h_{i1})LT(h_{i2}) \cdots LT(h_{it})).$$

Example 1. Consider a subset $H = \{h_1 = x^2\partial + 1, h_2 = 2x\partial - \partial, h_3 = x\partial\}$ of $A = K\langle x, \partial \mid \partial x = x\partial + 1 \rangle$ which is the first Weyl algebra. A monomial $m(H) \in K\langle H \rangle_A$ is

$$m(H) = h_1 h_2 = (x^2\partial + 1)(2x\partial - \partial) = 2x^3\partial^2 - x^2\partial^2 + 2x^2\partial + 2x\partial - \partial.$$

Further

$$\begin{aligned} LM(h_1)LM(h_2) &= x^3\partial^2 + x^2\partial \text{ and,} \\ \overline{LM}m(H) &= LM(LM(h_1)LM(h_2)) = x^3\partial^2. \end{aligned}$$

3. SAGBI Normal Form in G-Algebras

In this section, first, we define the process of reduction together with SAGBI normal form in G-algebras, following which we define the concept of SAGBI bases in G-algebra.

Definition 3. Let H and s be a subset and a polynomial in a G-algebra A , respectively. If there exists an H -monomial $m(H)$, and $k \in K$ satisfying $\overline{LT}(km(H)) = LT(s)$, then we say that

$$s_0 = s - km(H) \tag{1}$$

is a one-step s -reduction of s with respect to H . Otherwise, the s -reduction of s with respect to H is s itself. If we apply the one-step s -reduction process iteratively, we can achieve a special form of s with respect to H (which cannot be s -reduced further with respect to H), called SAGBI normal form, and write it as, $s_0 := \text{SNF}(s|H)$. For the reader's convenience, we give an algorithm for its computation.

Remark 1. During the reduction process inside the while loop, $LM(s_0)$ is strictly smaller than $LM(s)$ (by the choice of k and $m(H)$). Due to well-ordering of $>$, Algorithm 1 always terminates after a finite number of sweeps.

Algorithm 1 SNF($s \mid H$)

Require: $>$ a fixed well-ordering on the G-algebra A , $H \subset A$ and $s \in A$

Ensure: $h \in A$ the SAGBI normal form

```

 $s_0 := s$ 
 $H_{s_0} := \{km(H) \mid k \in K \text{ and } \overline{LT}(km(H)) = LT(s_0)\}$ 
while  $s_0 \neq 0$  and  $H_{s_0} \neq \emptyset$  do
    choose  $km(H) \in H_{s_0}$ 
     $s_0 := s_0 - km(H)$ 
     $H_{s_0} := \{km(H) \mid k \in K \text{ and } \overline{LT}(km(H)) = LT(s_0)\}$ 
return  $s_0$ ;

```

Remark 2. For different choices “ $km(H)$ ” in the algorithm above, the output of SNF may also be different.

Following is an example of the SAGBI normal form in an enveloping algebra. Tables 1 and 2 in the above example shows that the SNF of different choices are uncommon. In the next example (see Tables 3–5) we use second Weyl algebra with all possible choices of the while loop of Algorithm 1.

Example 2. Let $A = \mathbb{Q}\langle e, f, h \mid fe = ef - h, he = eh + 2e, hf = fh - 2f \rangle$. Let S be a subalgebra of A generated by $H = \{q_1, q_2, q_3\} = \{e^2, f, fh + f\}$ and $g = e^2fh + eh + f$, associated with degrevlex ordering (dp). For the computation of $SNF(g \mid H)$, we use Algorithm 1.

Table 1. First possible choice—Example 2.

Turn	h_i	H_{h_i}	Choose	h_{i+1}
$i = 0$	g	$\{q_1q_3, q_3q_1\}$	q_1q_3	$-e^2f + eh + f$
$i = 1$	h_1	$\{q_1q_2, q_2q_1\}$	q_1q_2	$eh + f$
$i = 2$	h_2	\emptyset		$SNF(g \mid H) = eh + f$

Table 2. Second possible choice—Example 2.

Turn	h_i	H_{h_i}	Choose	h_{i+1}
$i = 0$	g	$\{q_1q_3, q_3q_1\}$	q_3q_1	$-5e^2f + 2eh^2 + 13eh + 10e + f$
$i = 1$	h_1	$\{q_1q_2, q_2q_1\}$	q_1q_2	$2eh^2 + 13eh + 10e + f$
$i = 2$	h_2	\emptyset		$SNF(g \mid H) = 2eh^2 + 13eh + 10e + f$

Example 3. Let $A = \mathbb{Q}\langle x_1, x_2, \partial_1, \partial_2 \mid \partial_i x_i = x_i \partial_i + 1 \rangle$, and the subalgebra S in A generated by $H = \{p_1, p_2\} = \{x_2^2 \partial_1 - 1, x_1 \partial_2 + 1\}$, and a polynomial $g = x_1 \partial_1^2 x_2^4 \partial_2 + x_1 x_2^2 + x_2^2 \partial_1^3$, associated with degrevlex ordering (dp). For the computation of the $SNF(g \mid H)$, we use Algorithm 1.

Table 3. First possible choice—Example 3.

Turn	h_i	H_{h_i}	Choose	h_{i+1}
$i = 0$	g	$\{p_1^2 p_2, p_1 p_2 p_1, p_2 p_1^2\}$	$p_1^2 p_2$	$-\partial_1^2 x_2^4 - 2\partial_1 x_2^4 \partial_2 + \partial_1^3 x_2^2 + 2x_1 \partial_1 x_2^2 \partial_2 + x_1 x_2^2 + 2\partial_1 x_2^2 + 2x_2^2 \partial_2 - x_1 \partial_2 - 1$
$i = 1$	h_1	$\{p_1^2\}$	p_1^2	$-2\partial_1 x_2^4 \partial_2 + \partial_1^3 x_2^2 + 2x_1 \partial_1 x_2^2 \partial_2 + x_1 x_2^2 + 2x_2^2 \partial_2 - x_1 \partial_2$
$i = 2$	h_2	\emptyset		$SNF(g \mid H) = -2\partial_1 x_2^4 \partial_2 + \partial_1^3 x_2^2 + 2x_1 \partial_1 x_2^2 \partial_2 + x_1 x_2^2 + 2x_2^2 \partial_2 - x_1 \partial_2$

Table 4. Second possible choice—Example 3.

Turn	h_i	H_{h_i}	Choose	h_{i+1}
$i = 0$	g	$\{p_1^2 p_2, p_1 p_2 p_1, p_2 p_1^2\}$	$p_1 p_2 p_1$	$-2x_1 \partial_1^2 x_2^3 - \partial_1^2 x_2^4 - \partial_1 x_2^4 \partial_2 + \partial_1^3 x_2^2 + 2x_1 \partial_1 x_2^2 \partial_2 - 2\partial_1 x_2^3 + 2x_1 \partial_1 x_2 + x_1 x_2^2 + 2\partial_1 x_2^2 + x_2^2 \partial_2 - x_1 \partial_2 - 1$
$i = 1$	h_1	\emptyset		$-2x_1 \partial_1^2 x_2^3 - \partial_1^2 x_2^4 - \partial_1 x_2^4 \partial_2 + \partial_1^3 x_2^2 + 2x_1 \partial_1 x_2^2 \partial_2 - 2\partial_1 x_2^3 + 2x_1 \partial_1 x_2 + x_1 x_2^2 + 2\partial_1 x_2^2 + x_2^2 \partial_2 - x_1 \partial_2 - 1$

Table 5. Third possible choice—Example 3.

Turn	h_i	H_{h_i}	Choose	h_{i+1}
$i = 0$	g	$\{p_1^2 p_2, p_1 p_2 p_1, p_2 p_1^2\}$	$p_2 p_1^2$	$-4x_1 \partial_1^2 x_2^3 - \partial_1^2 x_2^4 + \partial_1^3 x_2^2 + 2x_1 \partial_1 x_2^2 \partial_2 + 4x_1 \partial_1 x_2 + x_1 x_2^2 + 2\partial_1 x_2^2 - x_1 \partial_2 - 1$
$i = 1$	h_1	\emptyset		$SNF(g H) = -4x_1 \partial_1^2 x_2^3 - \partial_1^2 x_2^4 + \partial_1^3 x_2^2 + 2x_1 \partial_1 x_2^2 \partial_2 + 4x_1 \partial_1 x_2 + x_1 x_2^2 + 2\partial_1 x_2^2 - x_1 \partial_2 - 1$

Let S be a subalgebra of G -algebra A and $H \subset S$. Our interest lies in the case when SAGBI normal form $s_0 = 0$ for $s \in S$. If there is at least one choice of H -monomials such that $s_0 = 0$, then we say s reduces weakly over H , and reduces strongly if all possible choices give $s_0 = 0$.

Definition 4. Let S be a subalgebra of G -algebra A . A subset $H \subset S$ is called a SAGBI basis for S if $\forall s \in S, s \neq 0, \exists$ a H -monomial, $m(H)$ in $K\langle H \rangle_A$ such that

$$LM(s) = \overline{LM}(m(H)) \tag{2}$$

The following proposition illustrates that $s \in K\langle H \rangle_A$ reduces strongly to $s_0 = 0$ if H is a SAGBI basis of S .

Proposition 2. Let S be subalgebra of A and $H \subseteq S$. We assume H to be a SAGBI basis of S , then

1. For each $s \in A, s \in S$ if and only $SNF(s|H) = 0$
2. H generates the subalgebra S i.e., $S = K\langle H \rangle_A$.

Proof. 1. First assume $SNF(s|H) = 0$, then $s = \sum k_i m_i(H)$ where $k_i \in K$ and hence $s \in S$. Conversely, suppose that $s \in S$ and $SNF(s|H) \neq 0$ then it cannot be reduced further i.e., $LM(SNF(s|H)) \neq \overline{LM}(m(H))$, for any H -monomial $m(H)$ and this contradicts that H is a SAGBI basis.

2. Follows from (1), $s \in S$ if and only if $SNF(s|H) = 0$, that is, $s = \sum k_i m_i(H)$ with $k_i \in K$, it implies $s \in K\langle H \rangle_A$. which shows $S = K\langle H \rangle_A$.

□

4. SAGBI Basis Construction in G -Algebras

For the computation of SAGBI bases in G -algebra, we propose an algorithm and explore some ingredients that are necessary for this construction. Throughout this section, let A be a G -algebra over the field K .

Definition 5. Let $H \subseteq A$ and $m(H)$ and $m'(H)$ be H -monomials. The pair $(m(H), m'(H))$ is a critical pair of a H if $\overline{LM}(m(H)) = \overline{LM}(m'(H))$. The T -polynomial of critical pair is defined as $T(m(H), m'(H)) = m(H) - km'(H)$ where $k \in K$ such that $\overline{LT}(m(H)) = \overline{LT}(m'(H))$.

Definition 6. Let H be a set of polynomials in A and $S = K\langle H \rangle_A$ be a subalgebra in A . We consider $P \in S$ with the representation $P = \sum_{i=1}^t k_i m_i(H)$. Then the height of P with respect to this representation is defined as $ht(P) = \max_{i=1}^t \{\overline{LM}(m_i(H))\}$, where the maximum is taken with respect to term ordering in A .

Remark 3. The height is defined for a specific representation of elements of A , not for the elements itself.

Theorem 1. (SAGBI Basis Criterion) Assume H generates S as a subalgebra in A , then H is a SAGBI basis of S if every T -polynomial of every critical pair of H gives zero SAGBI normal form.

Proof. Assume H is a SAGBI basis of S . Since every T -polynomial is an element of $S = K\langle H \rangle_A$, its SAGBI normal form is equal to zero by part (1) of Proposition 2.

Conversely, suppose given $0 \neq s \in S$. It is sufficient to prove that it has a representation $s = \sum_{p=1}^t k_p m_p(H)$, where $k_p \in K$ and $m_p(H) \in K\langle H \rangle_A$ with $LM(s) = ht(\sum_{p=1}^t k_p m_p(H))$.

Let $s \in S$ with representation $s = \sum_{p=1}^t k_p m_p(H)$ with smallest possible height X among all possible representations of s in S , that is $X = \max_{p=1}^t \{LM(m_p(H))\}$. Clearly $LM(s) \leq X$.

Suppose $LM(s) \leq X$ i.e., cancellation of terms occur then there exist at least two H -monomials such that their leading monomial is equal to X . Assume we have only two H -monomials $m_i(H), m_j(H)$ in the representation $s = \sum_{p=1}^t k_p m_p(H)$ such that $LM(m_i(H)) = LM(m_j(H)) = X$. If $T(m_i(H), m_j(H)) = m_i(H) - km_j(H)$, we can write

$$\begin{aligned}
 s &= \sum_{p=1}^t k_p m_p(H) \\
 &= k_i(m_i(H) - km_j(H)) + (k_j + k_i k)m_j(H) + \sum_{p=1, p \neq i, j}^t k_p m_p(H) \\
 &= k_i T(m_i(H), m_j(H)) + (k_j + k_i k)m_j(H) + \sum_{p=1, p \neq i, j}^t k_p m_p(H) \tag{3}
 \end{aligned}$$

Since $T(m_i(H), m_j(H))$ has a zero SAGBI normal form, then this T -polynomial is either zero or can be written as sum of H -monomials of height $LM(T(m_i(H), m_j(H)))$ which is less than X . If $k_j + k_i k$ is equal to zero, then the right-hand side of Equation (3) is a representation of s that has the height less than X , which contradicts our initial assumption that we had chosen a representation of s that had the smallest possible height. Otherwise, the height is preserved, but on the right-hand side of Equation (3), we have only one H -monomial $m_j(H)$ such that $LM(m_j(H)) = X$, which is a contradiction as at least two H -monomials of such type must exist in the representation of s . \square

Remark 4. The necessary critical pairs used in SAGBI basis testing are those critical pairs $((m(H), m'(H)))$ which cannot be factor as $m(H) = m_1(H) \cdots m_t(H)$, $m'(H) = m'_1(H) \cdots m'_t(H)$ with $LM m_i(H) = LM m'_i(H)$ for all i . The T -polynomial induced by a necessary critical pair is called the necessary T -polynomial. Since G -algebras are finite factorization domains (Theorem 1.3, [17]), therefore for any critical pair $((m(H), m'(H)))$ (possibly not a necessary critical pair), the H -monomials $m(H)$ and $m'(H)$ have finite irreducible factors. The necessary critical pairs will be formed by these irreducible factors, therefore the zero SAGBI normal form of T -polynomials induced by necessary critical pairs implies the SAGBI normal form of T -polynomial of a critical pair $((m(H), m'(H)))$, will be zero (for details, see proposition 6 of [14]).

Using Remark 4, Theorem 1 can be restated by replacing every critical pair with necessary critical pairs i.e., a set that generates a subalgebra in a G -algebra is a SAGBI basis if and only if the T -polynomial of all necessary critical pairs of that set gives zero SAGBI normal form.

The following example illustrates Remark 4.

Example 4. Let $A = \mathbb{Q}\langle x, \partial \mid \partial x = x\partial + 1 \rangle$ be the first Weyl algebra. Let $S \subseteq A$ be the subalgebra generated by $H = \{x^2, x\partial, \partial^2\}$ with $x >_{lex} \partial$. Let

$$\begin{aligned}
 m(H) &= x^3\partial + x^2\partial^3 + x^2\partial^2 + x\partial^4 + x\partial, \text{ and} \\
 m'(H) &= x^3\partial - x^2\partial^3 + x\partial^4 - \partial^6 + 3\partial^3.
 \end{aligned}$$

with $\overline{LM}m(H) = \overline{LM}m'(H)$ then $(m(H), m'(H))$ is not a necessary critical pair because they can be written in factored form as

$$\begin{aligned} m(H) &= (x^2 + x\partial)(x\partial + \partial^3) := m_1(H)m_2(H), \text{ and} \\ m'(H) &= (x^2 + \partial^3)(x\partial - \partial^3) := m'_1(H)m'_2(H), \end{aligned}$$

with $\overline{LM}m_i(H) = \overline{LM}m'_i(H)$ for $i = 1, 2$. Please note that $(m_i(H), m'_i(H))$ are necessary critical pairs. Also, observe that

$$\begin{aligned} T(m(H), m'(H)) &= 2x^2\partial^3 + x^2\partial^2 + x\partial + \partial^6 - 3\partial^3 \\ &= (x\partial - \partial^3)(x\partial + \partial^3) + (x^2 + \partial^3)(2\partial^3) \\ &= T(m_1(H), m'_1(H))m_2(H) + m'_1(H)T(m_2(H), m'_2(H)). \end{aligned}$$

Since SAGBI normal form of T -polynomials on the right-hand side reduces to zero, therefore the SAGBI normal form of $T(m(H), m'(H))$ also vanishes.

Now we give an algorithm based on the SAGBI Basis Criterion to compute SAGBI basis.

Proposition 3. Let $H_\infty = \cup H$, accumulated over a while loop in Algorithm 2. Then H_∞ is a SAGBI basis for $K \langle H_0 \rangle_A$. Furthermore, if $K \langle H_0 \rangle_A$ is a finitely generated subalgebra (i.e., H_0 is a finite set) and admits a finite SAGBI basis, then Algorithm 2 stops and yields a finite SAGBI basis for $K \langle H_0 \rangle_A$

Algorithm 2 SAGBI Construction Algorithm

Require: $>$ a fixed well-ordering on the G -algebra A , $H_0 \subseteq A$

Ensure: A SAGBI basis H for $K \langle H_0 \rangle_A$

$H = H_0$ and old $H = \emptyset$

while $H \neq \text{old } H_0$ **do**

Compute $C =$ set of all necessary critical pairs of H

$D = \{T(m(H), m'(H)) : (m(H), m'(H)) \in C\}$

$\text{Red} = \{\text{SNF}(p|H) \mid p \in D\} \setminus \{0\}$

old $H = H$

$H = H \cup \text{Red}$

return H ;

Proof. First, we will prove the correctness of Algorithm 2, despite its termination.

Correctness: Let $C_\infty = \cup C$ (accumulated over a while loop). We will show that for any arbitrary $(m(H_\infty), m'(H_\infty)) \in C_\infty$, for which T -polynomial $p = T(m(H_\infty), m'(H_\infty))$, we have $\text{SNF}(p|H_\infty) = 0$.

Since $m(H_\infty)$ and $m'(H_\infty)$ can always be written in terms of a finite number of elements, $h_i \in H_\infty$. Also, the sets H are nested, therefore these specific h_i 's necessarily be in H_{n_o} , which is formed during the execution of a finite number, n_o , of loops. We can assume that $m(H_\infty) = m(H_{n_o})$ and $m'(H_\infty) = m'(H_{n_o})$ which implies $p = T(m(H_{n_o}), m'(H_{n_o}))$. Clearly, either $\text{SNF}(p|H_{n_o}) = 0$ or $\text{SNF}(p|H_{n_o+1}) = 0$. This implies that $\text{SNF}(p|H_\infty) = 0$, thus, by Theorem 1, H_∞ is a SAGBI basis for $K \langle H_\infty \rangle_A = K \langle H_0 \rangle_A$.

Termination: Now, we suppose that $K \langle H_0 \rangle_A$ has a finite SAGBI basis S . Because H_∞ is also a SAGBI basis for $K \langle H_0 \rangle_A$, then for each $s \in S$, we have the following expression $LM(s) = \overline{LM}(m(H_\infty))$ for

some H_∞ -monomial $m(H_\infty)$.

These H_∞ -monomials are in terms of finitely many elements of H_∞ , we represent this set by \widehat{H} . Please note that \widehat{H} is a finite set and $m(H_\infty) = m(\widehat{H})$. Observe that $\overline{LM}(m(\widehat{H})) = \overline{LM}(m(H_\infty)) = LM(s)$ which implies \widehat{H} is a SAGBI basis of $K\langle H_0 \rangle_A$. The finite set \widehat{H} must be a subset of H_{n_0} which is produced after a finite number n_0 of loops. Therefore, the set H_{n_0} is a SAGBI basis of $K\langle H_0 \rangle_A$ and by Theorem 1 the algorithm will terminate after the next pass.

Now we will prove that H_{n_0} is finite for any finite input H_0 . It follows from Remark 4 that for a finite set $H \subset A$, there exists finitely many irreducible pairs of H -monomials $m(H), m'(H)$ such that $\overline{LM}(m(H)) = \overline{LM}(m'(H))$. This implies that there exist finitely many necessary critical pairs at each step in Algorithm 2, i.e., the set C after the while loop is finite at each step, therefore the output of the while loop should necessarily be finite. Hence starting with a finite set H_0 in Algorithm 2 and completing a strictly finite number of loops n_0 , each loop produces a finite output. We finally achieve the output H_{n_0} which is a finite SAGBI basis. \square

We now give examples of SAGBI bases.

Example 5. Let $A = \mathbb{Q}\langle x, \partial \mid \partial x = x\partial + 1 \rangle$ be the first Weyl algebra. Let $S \subseteq A$ be the subalgebra generated by $H = \{p_1 = x^2, p_2 = x\partial, p_3 = \partial^2\}$ with $x >_{\text{lex}} \partial$. Then its necessary critical pairs are $(p_1 p_3, p_2^2), (p_1 p_3, p_3 p_1)$ and $(p_1 p_3, p_3 p_1)$ gives T -polynomials that are reduced to zero. Hence $H = \{p_1 = x^2, p_2 = x\partial, p_3 = \partial^2\}$ is a SAGBI basis.

In the next example we add some elements to the generating set during the construction of SAGBI basis.

Example 6. Let $A = \mathbb{Q}\langle e, f, h \mid fe = ef - h, he = eh + 2e, hf = fh - 2f \rangle$ be an enveloping algebra. Let $S \subseteq A$ be the subalgebra generated by $H = \{e, h^2\}$. We construct SAGBI basis of S with respect to the lex ordering.

Let $p_1 = e, p_2 = h^2$, then for the necessary critical pair $(m_1(H), m_2(H))$ where

$$\begin{aligned} m_1(H) &= p_2 p_1 = eh^2 + 4eh + 4e, \text{ and} \\ m_2(H) &= p_1 p_2 = eh^2, \end{aligned}$$

the T -polynomial is $T(m_1(H), m_2(H)) = m_1(H) - m_2(H) = 4eh + 4e$. It is not reduced by elements of H , so $p_3 = eh + e$ and $H = \{p_1, p_2, p_3\}$. For the necessary critical pair $(m_3(H), m_4(H))$ where

$$\begin{aligned} m_3(H) &= p_3^3 = e^2 h^2 + 4e^2 h + 3e^2, \text{ and} \\ m_4(H) &= p_1^2 p_2 = e^2 h^2, \end{aligned}$$

the T -polynomial is

$$T(m_3(H), m_4(H)) = m_3(H) - m_4(H) = 4e^2 h + 3e^2 = 4p_1 p_3 + 3p_1^2 := g,$$

and $\text{SNF}(g \mid H) = 0$ and all T -polynomials of necessary critical pairs give zero SAGBI normal form. Hence $H = \{e, h^2, eh + e\}$ is a SAGBI basis.

The next example shows that similar to the commutative case, a SAGBI basis of a subalgebra could be infinite.

Example 7. (Infinite SAGBI basis in the enveloping algebra)

Let $A = \mathbb{Q}\langle e, f, h \mid fe = ef - h, he = eh + 2e, hf = fh - 2f \rangle$. Let $S \subseteq A$ be the subalgebra generated by $H = \{p_1 = h, p_2 = e^2, p_3 = f^2, p_4 = efh\}$. We construct SAGBI basis of S with respect to the lex ordering.

For the necessary critical pair $(m_1(H), m_2(H))$ where

$$\begin{aligned} m_1(H) &= p_2 p_3 = e^2 f^2, \text{ and} \\ m_2(H) &= p_3 p_2 = e^2 f^2 - 4efh + eh^2 + h^2 + 2h, \end{aligned}$$

the T -polynomial is

$$T(m_1(H), m_2(H)) = m_1(H) - m_2(H) = 4efh - eh^2 + h^2 + 2h =: g_1,$$

and $\text{SNF}(g_1 \mid H) = eh^2 - h^2 - 2h =: p_5$. It is not reduced by elements of H , so $H = \{p_1, p_2, p_3, p_4, p_5\}$. Continuing in this way we get an infinite SAGBI basis $H = \{h, e^2, f^2, efh, eh^2, ef^2, fh^2, \dots\}$.

In this paper, we develop the theory of SAGBI bases in G -Algebras and its corresponding algorithms. It is useful to understand the structure of subalgebras in a given G -algebra. The theory of Gröbner bases of ideals of a subalgebra in a polynomial ring, termed as SAGBI-Gröbner basis was developed by Miller [18]. This work can be evolved into the theory of SAGBI-Gröbner bases in G -algebras, which illustrate a better significance of ideals in a given subalgebra of a G -algebra.

Author Contributions: All authors contributed equally.

Funding: The APC was funded by Institute of Business Administration, Karachi, Pakistan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Levandovskyy, V. Non-Commutative Computer Algebra for Polynomial Algebras: Gröbner Bases, Applications and Implementation. Ph.D. Thesis, Universität Kaiserslautern, Kaiserslautern, Germany, 2005.
2. Dixmier, J. *Enveloping Algebras*; Akademie-Verlag: New York, NY, USA, 1977.
3. Apel, J. Gröbnerbasen in Nichtkommutativen Algebren und ihre Anwendung. Ph.D. Thesis, Universität Leipzig, Leipzig, Germany, 1988.
4. Mora, T. Groebner bases in non-commutative algebras. In *International Symposium on Symbolic and Algebraic Computation*; Springer: Berlin/Heidelberg, Germany, 1988; pp. 150–161.
5. Kandri-Rody, A.; Weispfenning, V. Non-commutative Gröbner bases in algebras of solvable type. *J. Symb. Comput.* **1990**, *9*, 1–26. [[CrossRef](#)]
6. Kredel, H. *Solvable Polynomial Rings*; Shaker: Aachen, Germany, 1993.
7. Li, H. *Noncommutative Gröbner Bases and Filtered-Graded Transfer*; Lecture Notes in Mathematics; Springer: Cham, Switzerland, 2002; Volume 1795.
8. Bueso, J.L.; Gómez-Torrecillas, J.; Verschoren, A. *Algorithmic Methods in Non-Commutative Algebra: Applications to Quantum Groups*; Springer: Dordrecht, The Netherlands, 2013.
9. Rosenberg, A.L. *Noncommutative Algebraic Geometry and Representations of Quantized Algebras, Mathematics and Its Applications*; Springer: Dordrecht, The Netherlands, 1995; Volume 330.
10. Gordan, P. *Les Invariants Des Formes Binaires*; Gauthier-Villars: San Francisco, CA, USA, 1900.
11. Buchberger, B. Ein Algorithmus zum Au ndedder Basiselemente des Restklassenringes nach einem Nulldimensionalen Polynomideal. Ph.D. Thesis, University of Innsbruck, Innsbruck, Austria, 1965.
12. Robbiano, L.; Sweedler, M. Subalgebra bases. In *Commutative Algebra*; Lecture Notes in Mathematics; Springer: Berlin/Heidelberg, Germany, 1990; Volume 1430, pp. 61–87.
13. Kapur, D.; Madlener, K. A Completion Procedure for Computing a Canonical Basis for a k -Subalgebra. In *Computers and Mathematics*; Springer: New York, NY, USA, 1989; pp. 1–11.

14. Nordbeck, P. Canonical subalgebraic bases in non-commutative polynomial rings. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*; ACM: New York, NY, USA, 1998; pp. 140–146.
15. Greuel, G.-M.; Pfister, G.; Schönemann, H. Singular 4-1-1—A Computer Algebra System for Polynomial Computations. Available online: <http://www.singular.uni-kl.de> (accessed on 12 February 2019).
16. Greuel, G.-M.; Pfister, G. *A Singular Introduction to Commutative Algebra. With Contributions by Olaf Bachmann, Christoph Lossen and Hans Schönemann*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2007.
17. Bell, J.; Heinle, A.; Levandovskyy, V. On noncommutative finite factorization domains. *Trans. Am. Math. Soc.* **2017**, *369*, 2675–2695. [[CrossRef](#)]
18. Miller, J.L. *Effective Algorithms for Intrinsically Computing SA GBI-Groebner Bases in a Polynomial Ring over a Field*; London Mathematical Society Lecture Note Series; Cambridge University Press: Cambridge, UK, 1998; pp. 421–433.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).