Computer Science Journal of Moldova, vol.**8**, no.1(22), 2000

# On the computation of Hilbert series and Poincaré series for algebras with infinite Gröbner bases

Jonas Mánsson

**Abstract**

In this paper we present algorithms to compute finite state automata which, given any rational language, recognize the languages of normal words and $n$-chains. We also show how these automata can be used to compute the Hilbert series and Poincaré series for any algebra with a rational set of leading words of its minimal Gröbner basis.

## 1 Introduction

The computation of Hilbert series and Poincaré series for commutative algebras is a well-studied problem and many efficient algorithms have been developed. For noncommutative algebras the situation is substantially different and the problem is in general very hard. First of all the equality problem for words is generally not solvable. The unsolvability shows already for relatively simple examples (see for instance [6]). In addition, the rationality of Hilbert series is not guaranteed even for finitely presented algebras (see [5]). Nevertheless, for many important classes of algebras, there do exist algorithms for computing Hilbert series and Poincaré series, e.g algebras with finite Gröbner bases. In [7] the notion of automaton algebras was introduced, with the property of having a rational set of normal words. It turns out that this class of algebras always admits rational Hilbert series. Moreover, provided the set of normal words can be recognized by a finite state automaton, there is an algorithm to compute it. The computation of Poincaré

series can be carried out by investigating an appropriate free resolution. In [1] Anick constructed such a resolution by means of so called $n$-chains, a kind of generalized obstructions created from the leading words of the corresponding Gröbner basis. By presenting the $n$-chains as a graph the Poincaré series can be obtained in a similar way as for Hilbert series.

For both these matters it is essential to in some way convert the information given by the Gröbner basis to a rational expression for the set of normal words and $n$-chains respectively. In case of finite Gröbner bases this is possible. The details can be found in [7]. In this paper we will describe algorithms for computing finite state automata recognizing the set of normal words and $n$-chains, given an infinite Gröbner basis with a rational set of leading words. In particular, the construction of the automaton for $n$-chains is an entirely new result, making it possible to effectively compute the monomial Poincaré series for any given automaton algebra. For normal words we also prove that the provided algorithm is optimal in terms of time complexity. Some examples of how to use this technique for monomial subalgebras with a rational set of generators will also be given. Finally some computational aspects of the computation of Hilbert series and Poincaré series will be discussed.

## 2 Definitions and notions

By an *alphabet* $X = \{x_1, x_2, \ldots, x_m\}$ we mean a finite set of *letters*. We will always assume that the elements of $X$ are graded, i.e there is a mapping $\deg : X \longrightarrow \mathbb{Z}^+$. For simplicity we will in all our examples assume that $\deg x = 1$ for all $x \in X$. A *language* $\mathcal{L}$ is a collection of finite *words* of letters in $X$. The language of all words (including the empty word $\mathbf{1}$) will be denoted by $X^*$. In algebraic notation this corresponds to the free monoid generated by $X$. The grading on $X$ naturally extends to $X^*$. For any language $\mathcal{L}$ we denote by $\mathcal{L}^*$ the set of words consisting of all finite products of elements in $\mathcal{L}$ (including the empty word $\mathbf{1}$).

For any field $K$ we let $K\langle X\rangle$ be the free associative algebra gener-

ated by $X$. From now on it is assumed that there is given an *admissible* order on $X^*$, i.e a total order preserved under monoid multiplication: $f < g \Rightarrow hfk < hgk$ for all $f, g, h, k \in X^*$. In all our examples we will use a so called degree-lexicographical order

$$f < g \Leftrightarrow \begin{cases} \deg f < \deg g \\ \text{or if } \deg f = \deg g,\ f \text{ is smaller than } g \\ \quad \text{lexicographically} \end{cases}$$

Having introduced such an order we can to every element $f \in K\langle X\rangle$ associate its leading word $\hat{f} \in X^*$. For any subset $F \subseteq K\langle X\rangle$ we define the language $\hat{F}$ of leading words as $\hat{F} = \{\hat{f} | f \in F\}$. For any $u, v \in X^*$, if $u$ is a subword of $v$ we say that $u$ divides $v$ and write $u|v$.

**Definition 2.1 (Gröbner basis)** *Let $I$ be a two-sided ideal in $K\langle X\rangle$. A subset $G \subseteq I$ is a* Gröbner basis *for $I$ if, for all $f \in I$, there exists $g \in G$ such that $\hat{g}|\hat{f}$.*

We specially point out that such a basis always exists, e.g the ideal itself. It is well-known that there exists a process to compute a Gröbner basis starting from some finite set of relations (see e.g [9]). On the other hand the result may be infinite, which creates problems in terms of computer calculation. Later we will return to this problem and suggest ways to overcome it in some cases.

A word $u \in X^*$ is called *normal* modulo an ideal $I$, if for every $v \in \hat{G}$, $v \not| u$. In analogy with this definition we say that $u \in X^*$ is normal with respect to a language $\mathcal{L}$ if, for every $v \in \mathcal{L}$, $v$ does not divide $u$. A language $\mathcal{L}$ is called *reduced* if no word in $\mathcal{L}$ is a subword of another. In particular a Gröbner basis $G$ is minimal (i.e no smaller subset is a Gröbner basis) if $\hat{G}$ is a reduced language. With every language $\mathcal{L}$ we can associate its unique reduced language $\mathcal{L}_R$ being the largest subset of $\mathcal{L}$ the elements of which do not contain any other subword in $\mathcal{L}$.

In what follows we will also come across the notion of $n$-chains (and their tails):

**Definition 2.2 (n-chain)** *Let $\mathcal{L}$ be a reduced language. The notion of* $n$-chain *is defined by induction on $n$. All $x \in X$ are* 0-chains *and*

*all coincide with their* tails. *An* $n$-chain *is a word* $f = gt$ *where* $g$ *is an* $(n-1)$*-chain and* $t$ *a normal word such that, if* $r$ *is the tail of* $g$*,* $rt$ *contains a unique element in* $\mathcal{L}$ *and this unique occurrence is the ending of the word* $rt$*. The* tail *of* $f$ *is defined to be* $t$*. The language of* $n$*-chains is denoted* $C_n(\mathcal{L})$*.*

Note that from the definition it follows that no $n$-chain is a proper subword of another $n$-chain.

# 3 Rational languages and automata

In this section we will provide algorithms for computing various languages connected with different structures of a given algebra. Our main object of interest will be rational languages, i.e languages recognized by finite state automata.

## 3.1 Deterministic and nondeterministic automata

We recall the definition of a finite state automaton.

**Definition 3.1** *A* finite state automaton *(FSA)* $\mathcal{M} = \{Q, X, q_1, F\}$ *consists of the following five objects:*

- *A finite, nonempty set* $Q$*. The elements of* $Q$ *are called* states*;*
- *A finite, nonempty alphabet* $X$*.*
- *A unique state* $q_1 \in Q$ *called the* initial state *of* $\mathcal{M}$*;*
- *For each* $x \in X$ *a* transition function $\mathcal{M}_x : Q \longrightarrow \mathcal{P}(Q)$*, where* $\mathcal{P}(Q)$ *is the set of all subsets of* $Q$*;*
- *A nonempty subset* $F$ *of* $Q$*. The elements of* $F$ *are called* terminal states*;*

$\mathcal{M}$ *is a* deterministic *finite state automaton (DFSA) if, for all* $q \in Q$ *and* $x \in X$, $\mathcal{M}_x(q)$ *never contains more than one element*[1]. *Otherwise it is called* nondeterministic *(NDFSA).*

Every FSA $\mathcal{M}$ can be represented by a graph $\mathcal{G}(\mathcal{M})$, the states being the vertices and the transition functions $\mathcal{M}_x$ defining the edges. Depending on the situation we will alternately refer to $\mathcal{M}$ as its abstract definition and alternately as its graph.

To represent the way in which a FSA operates for any given input we extend the set of transition functions to $\mathcal{M}_u : \mathcal{P}(Q) \longrightarrow \mathcal{P}(Q),\ u \in X^*$. This class of functions is defined recursively by

$$\begin{cases} \mathcal{M}_{\mathbf{1}}(r) = r & r \in \mathcal{P}(Q) \\ \mathcal{M}_x(r) = \cup_{q \in r} \mathcal{M}_x(q) & x \in X, r \in \mathcal{P}(Q) \\ \mathcal{M}_{uv}(r) = (\mathcal{M}_v \circ \mathcal{M}_u)(r) & u, v \in X^*, r \in \mathcal{P}(Q) \end{cases}$$

In other words, given a word $u \in X^*$ and a set of states $r$, $\mathcal{M}_u(r)$ returns the subset of states reached from $r$ by all possible combinations of traversions induced by $u=$ in the corresponding graph $\mathcal{G}(\mathcal{M})$. We now define

$$u \in \mathcal{L}(\mathcal{M}) \Leftrightarrow \mathcal{M}_u(q_1) \cap F \neq \emptyset$$

and $\mathcal{L}(\mathcal{M})$ is referred to as the language *recognized* by $\mathcal{M}$. For convenient reasons we will write any subset of $Q$ consisting of a single element without brackets.

According to the definition it may seem that the concept of nondeterminism is very powerful, but surprisingly it is not. The following proposition is well-known in the theory of automata (see for instance [3]). We have also included the proof in order to demonstrate the kind of technique we will use later on.

**Proposition 3.1** *Let* $\mathcal{N}$ *be a NDFSA recognizing a language* $\mathcal{L}$. *Then there is an algorithm to compute a DFSA* $\mathcal{M}$ *recognizing the same language.*

[1] A more common definition of deterministic automata requires $\mathcal{M}_x(q)$ to contain exactly one element. However, allowing empty transitions does not turn $\mathcal{M}$ into a really nondeterministic machine, since any undefined transition can be considered as a transition to a "garbage state", i.e an additional nonterminal state which loops to itself for all $x \in X$.

**Proof** Let $\mathcal{N} = \{Q, X, q_1, F\}$ and let $\mathcal{M} = \{Q', X, q_1', F'\}$ be the DFSA defined by

$$\begin{cases} Q' = \{q_r' | r \in \mathcal{P}(Q)\} & \\ \text{initial state } q_1' = q_{q_1}' & \\ \mathcal{M}_x(q_r') = q_{\mathcal{N}_x(r)}' & r \in \mathcal{P}(Q),\ x \in X \\ q_r' \in F' \Leftrightarrow r \cap F \neq \emptyset & r \in \mathcal{P}(Q) \end{cases}$$

To prove that $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{N})$ is immediate from the construction of $\mathcal{M}$

$$u \in \mathcal{L}(\mathcal{M}) \Leftrightarrow F' \ni \mathcal{M}_u(q_1') = \mathcal{M}_u(q_{q_1}') = q_{\mathcal{N}_u(q_1)}' \Leftrightarrow$$

$$\Leftrightarrow \mathcal{N}_u(q_1) \cap F \neq \emptyset \Leftrightarrow u \in \mathcal{L}(\mathcal{N}) \quad \square$$

Indexing the new states with $\mathcal{P}(Q)$, the time complexity is clearly exponential in $n$. However, for the majority of natural instances we will encounter the average complexity will be much lower. Later we will focus on some strategies to improve the average complexity for particular instances.

### 3.2 Computation of some important rational languages

Now we are ready to prove the following proposition:

**Proposition 3.2** *Given a DFSA $\mathcal{M}$ which recognizes the language $\mathcal{L}$, there is an algorithm to compute a DFSA which recognizes*

*a) the language of normal words with respect to $\mathcal{L}(\mathcal{M})$.*
*b) the monoid generated by $\mathcal{L}(\mathcal{M})$.*
*c) the reduced language $\mathcal{L}(\mathcal{M})_R$*
*d) the n-chains of $\mathcal{L}(\mathcal{M})$.*

**Proof** a) In order to find an automaton for the language of normal words we first create one for the complement language, i.e for the two-sided monoid ideal generated by $\mathcal{L}(\mathcal{M})$. For this we modify $\mathcal{M} = \{Q, X, q_1, F\}$ into a NDFSA $\mathcal{N} = \{Q, X, q_1, F\}$ by letting $\mathcal{N}_x(q) = \mathcal{M}_x(q) \cup q$ for all $q \in F \cup q_1$ and $x \in X$. All other transitions remain

unchanged. To prove that $\mathcal{L}(\mathcal{N})$ is the complement language, first assume $u$ is not normal. Then there exists $v \in \mathcal{L}(\mathcal{M})$ such that $u = u_1 v u_2$ for some words $u_1, u_2 \in X^*$. Now since $q_1 \in \mathcal{N}_u(q_1)$ we get

$$F \ni \mathcal{M}_v(q_1) \subseteq (\mathcal{M}_v \circ \mathcal{N}_{u_1})(q_1) \subseteq (\mathcal{N}_v \circ \mathcal{N}_{u_1})(q_1)$$

and consequently $(\mathcal{N}_v \circ \mathcal{N}_{u_1})(q_1) \cap F \neq \emptyset$. From the construction of $\mathcal{N}$ it is clear that for any $r \in \mathcal{P}(Q)$ with $r \cap F \neq \emptyset$ and $u \in X^*$ we have $\mathcal{N}_u(r) \cap F \neq \emptyset$. In particular

$$\mathcal{N}_u(q_1) \cap F = (\mathcal{N}_{u_2} \circ (\mathcal{N}_v \circ \mathcal{N}_{u_1}))(q_1) \cap F \neq \emptyset$$

which means $u \in \mathcal{L}(\mathcal{N})$.

For the opposite inclusion let $u \in \mathcal{L}(\mathcal{N})$ and let $v$ be a left subword of $u$ such that $v \in \mathcal{L}(\mathcal{N})$ and does not contain any non-trivial left subword in $\mathcal{L}(\mathcal{N})$. We observe that $\mathcal{N}_v(q_1) = \cup_{v=t't}\mathcal{M}_t(q_1)$, so for at least one right subword $t$ of $v$ we must have $\mathcal{M}_t(q_1) \in F$. This in turn implies $t \in \mathcal{L}(\mathcal{M})$ and since $t$ in particular is a subword of $u$, $u$ is not normal.

According to proposition 3.1 it is algorithmic to compute a DFSA recognizing the same language. A DFSA for the language of normal words is now obtained by interchanging terminal and nonterminal states.

b) We can safely assume that $\mathbf{1} \in \mathcal{L}(\mathcal{M})$. Just like in a) we modify $\mathcal{M} = \{Q, X, q_1, F\}$ into a NDFSA $\mathcal{N} = \{Q, X, q_1, F\}$ by letting $\mathcal{N}_x(q) = \mathcal{M}_x(q) \cup \mathcal{M}_x(q_1)$ for all $q \in F$ and $x \in X$, leaving all other transitions unchanged. It remains to prove $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{M})^*$. Let $u \in \mathcal{L}(\mathcal{M})^*$ and let $u = u_1 u_2 \cdots u_n$ be some factorization with elements in $\mathcal{L}(\mathcal{M})$. We will prove $u \in \mathcal{L}(\mathcal{N})$ by induction on $n$. It is easily seen that $\mathcal{N}_{u_1}(q_1) \cap F \neq \emptyset$, so assume $\mathcal{N}_{u_1 u_2 \cdots u_{n-1}}(q_1) \cap F \neq \emptyset$. ¿From the construction of $\mathcal{N}$ we notice that $\mathcal{N}_{u_n}(q_1) \subseteq (\mathcal{N}_{u_n} \circ \mathcal{N}_{u_1 u_2 \cdots u_{n-1}})(q_1)$ and thus

$$\mathcal{M}_{u_n}(q_1) \subseteq \mathcal{N}_{u_n}(q_1) \subseteq (\mathcal{N}_{u_n} \circ \mathcal{N}_{u_1 u_2 \cdots u_{n-1}})(q_1) = \mathcal{N}_{u_1 u_2 \cdots u_n}(q_1)$$

Since $u_n \in \mathcal{L}(\mathcal{M})$

$$\emptyset \neq \mathcal{M}_{u_n}(q_1) \cap F \subseteq \mathcal{N}_{u_1 u_2 \cdots u_n}(q_1) \cap F$$

We have proved $\mathcal{N}_u \cap F \neq \emptyset$, or equivalently $u \in \mathcal{L}(\mathcal{N})$.

For the opposite inclusion let $u \in \mathcal{L}(\mathcal{N})$. It is easy to realize that we always can find a right subword $v \in \mathcal{L}(\mathcal{M})$ of $u$ and we write $u = wv$ for some $w \in X^*$. Moreover, since the transition functions for $\mathcal{M}$ and $\mathcal{N}$ only differ for terminal states, the definition of $\mathcal{N}$ enables us to choose $v$ so that either $w \in \mathcal{L}(\mathcal{N})$ or $w = \mathbf{1}$. If $w == \mathbf{1}$ we are done, otherwise an inductive argument proves the claim. By proposition 3.1 there is a DFSA recognizing the same language.

c) Consider $\mathcal{M} = \{Q, X, q_1, F\}$. Wishing to compute the corresponding reduced language we can safely make the assumption that $\mathcal{M}_x(q) = q$ for all $q \in F$ and $x \in X$. We modify $\mathcal{M}$ into a NDFSA $\mathcal{N} = \{Q, X, q_1, F\}$ by letting $\mathcal{N}_x(q_1) = \mathcal{M}_x(q_1) \cup q_1$ for all $x \in X$, leaving all other transitions unchanged. Let $X = \{x_1, x_2, \ldots, x_m\}$ and construct a DFSA $\mathcal{M}' = \{Q', X, q_1', F'\}$ in the following way:

$$\begin{cases} Q' = q_1' \cup (\cup_{1 \leq i \leq m} Q^i),\ Q^i = \{q^i_{(q,r)} | (q,r) \in Q \times \mathcal{P}(Q)\} \\ \text{initial state } q_1' \\ \mathcal{M}'_{x_i}(q_1') = q^i_{(\mathcal{M}_{x_i}(q_1), q_1)} \\ \mathcal{M}'_x(q^i_{(q,r)}) = \begin{cases} q^i_{(\mathcal{M}_x(q), \mathcal{N}_x(r))} & \text{if } q^i_{(q,r)} \notin F' \\ \emptyset & \text{otherwise.} \end{cases} \\ q^i_{(q,r)} \in F' \Leftrightarrow q \in F \end{cases}$$

He we use the convention that in case one of the indices in $q^i_{(q,r)}$ is empty, then $q^i_{(q,r)} = \emptyset$. By construction $\mathcal{L}(\mathcal{M}') \subseteq \mathcal{L}(\mathcal{M})$, and any $u \in \mathcal{L}(\mathcal{M})$ belongs to $\mathcal{L}(\mathcal{M}')$ if and only if it does not contain a proper left subword in $\mathcal{L}(\mathcal{M})$. Now we modify $\mathcal{M}'$ into a new DFSA $\mathcal{M}'' = \{Q', X, q_1', F''\}$ by letting $F'' = F' \cap \{q^i_{(q,r)} | r \cap F = \emptyset\}$. We want to prove $\mathcal{L}(\mathcal{M}'') = \mathcal{L}(\mathcal{M})_R$. Let $u = x_i v \in \mathcal{L}(\mathcal{M})_R \subseteq \mathcal{L}(\mathcal{M}')$ for some $x_i \in X$ so that $\mathcal{M}'_u(q_1') = q^i_{(q,r)} \in F'$. Suppose $r \cap F \neq \emptyset$. It is easily verified that $r = \cup_{v=t't} \mathcal{M}_t(q_1)$, and thus $t \in \mathcal{L}(\mathcal{M})$ for some right subword $t$ of $v$, which contradicts $u \in \mathcal{L}(\mathcal{M})_R$. It only remains $r \cap F = \emptyset$ and consequently $u \in \mathcal{L}(\mathcal{M}'')$.

For the opposite inclusion let $u \in \mathcal{L}(\mathcal{M}'')$. Trivially $u \in \mathcal{L}(\mathcal{M})$ since $\mathcal{L}(\mathcal{M}'') \subseteq \mathcal{L}(\mathcal{M}') \subseteq \mathcal{L}(\mathcal{M})$ so assume $u \notin \mathcal{L}(\mathcal{M})_R$. Then, either $u$

contains a proper left subword in $\mathcal{L}(\mathcal{M})$, or $u = x_i v$ for some letter $x_i$ and $v$ contains a subword in $\mathcal{L}(\mathcal{M})$. The first case is impossible since it would imply $u \notin \mathcal{L}(\mathcal{M}')$. As for the second case we note that the assumption $\mathcal{M}_x(q) = q$ for all $q \in F$ implies that $v$ contains a right subword $t \in \mathcal{L}(\mathcal{M})$. If $\mathcal{M}''_u(q'_1) = q^i_{(q,r)}$ a similar argument as before yields $r \cap F \neq \emptyset$, which contradicts $u \in \mathcal{L}(\mathcal{M}'')$.

d) According to c) there is an algorithm to compute a DFSA for $\mathcal{L}(\mathcal{M})_R$, which means we can assume $\mathcal{L}(\mathcal{M})$ is reduced. The strategy used here is in some way similar to the one in c). Indexing our new automaton with the product set $\mathcal{P}(Q) \times \mathcal{P}(Q)$, we can keep track of all beginnings of elements in $\mathcal{L}(\mathcal{M})$ in the current tail as well as recognize new such elements starting in the previous tail.

First we modify $\mathcal{M} = \{Q, X, q_1, F\}$ into a NDFSA $\mathcal{N} = \{Q, X, q_1, F\}$ by letting $\mathcal{N}_x(q_1) = \mathcal{M}_x(q_1) \cup q_1$ for all $x \in X$, leaving all other transitions unchanged. Let $\mathcal{M}' = \{Q', X, q'_1, F'\}$ be the DFSA defined by

$$\begin{cases} Q' = \{q'_{(r,s)} | (r,s) \in \mathcal{P}(Q) \times \mathcal{P}(Q)\} \\ \text{initial state } q'_1 = q'_{(q_1,q_1)} \\ \mathcal{M}'_x(q'_{(r,s)}) = q'_{(r',s')} \\ \qquad \text{where } (r',s') = (\mathcal{M}_x(s \setminus q_1), \mathcal{N}_x(q_1)) \text{ if } q'_{(r,s)} \in F' \\ \text{and } (r',s') = (\mathcal{M}_x(r), \mathcal{N}_x(s)) \text{ otherwise.} \\ q'_{(r,s)} \in F' \Leftrightarrow r \cap F \neq \emptyset \end{cases}$$

Just like in the proof of c), any empty index in $q^i_{(r,s)}$ implies $q^i_{(r,s)} = \emptyset$. Now $\mathcal{L}(\mathcal{M}')$ recognizes the language of $n$-chains of $\mathcal{L}(\mathcal{M})$ for all $n \geq 1$.

To prove this let $n \geq 1$. We prove that every $n$-chain belongs to $\mathcal{L}(\mathcal{M}')$ by induction. All 1-chains trivially belong to $\mathcal{L}(\mathcal{M}')$ (recall that $\mathcal{L}(\mathcal{M})$ is reduced) and we make the following induction assumption for $m < n$:

- If $u = vt$ is an $m$-chain for some $(m-1)$-chain $v$ with tail $t$ then $u \in \mathcal{L}(\mathcal{M}')$.
- In addition if $\mathcal{M}'_u(q'_1) = q'_{(r,s)}$ we have $s = \cup_{t=t''t'} \mathcal{M}_{t'}(q_1)$.

As for the second part of the assumption, notice that $\mathcal{M}_{t'}(q_1) \neq q_1$ for all right subwords $t'$ except the empty word. This follows from $\mathcal{L}(\mathcal{M})$ being reduced, since $\mathcal{M}$ cannot have any transitions to the initial state. Also note that for $m = 1$ we actually have $s = \mathcal{M}_u(q_1) \cup (\cup_{t=t''t'} \mathcal{M}_{t'}(q_1))$, as a consequence of not including 0-chains. However, this will not affect the validity of the proof, since $\mathcal{M}_u(q_1)$ is a terminal state and does not have any transitions leaving from it ($\mathcal{L}(\mathcal{M})$ is reduced).

Let $u$ be a $n$-chain ($n \geq 2$) with $u = vt_1t_2$, such that $v$ is a $(n-2)$-chain with tail $t_1$ and $vt_1$ an $(n-1)$-chain with tail $t_2$. By assumption $vt_1 \in \mathcal{L}(\mathcal{M}')$ and if $\mathcal{M}'_{vt_1}(q'_1) = q'_{(r,s)}$ we have $s = \cup_{t_1=t''t'} \mathcal{M}_{t'}(q_1)$. Assume $g$ is the smallest nonempty left subword of $t_2$ such that $vt_1g \in \mathcal{L}(\mathcal{M}')$. Note that such a smallest word always exists. It means

$$\mathcal{M}'_{vt_1g}(q'_1) = (\mathcal{M}'_g \circ \mathcal{M}'_{vt_1})(q'_1) = \mathcal{M}'_g(q'_{(r,s)}) = q'_{(r',s')}$$

with

$$\begin{cases} r' = \mathcal{M}_g \circ (\cup_{t_1=t''t', t'' \neq t_1} \mathcal{M}_{t'}(q_1)) = \\ \qquad = \cup_{t_1=t''t', t'' \neq t_1} \mathcal{M}_{t'g}(q_1) \\ s' = \cup_{g=t''t'} \mathcal{M}_{t'}(q_1) \end{cases}$$

Since $vt_1g \in \mathcal{L}(\mathcal{M}')$ we have $r' \cap F \neq \emptyset$ so $t'g \in \mathcal{L}(\mathcal{M})$ for some nonempty right subword $t'$ of $t_1$. If $g$ is a proper left subword of $t_2$ then $t_1t_2$ must contain at least two different elements in $\mathcal{L}(\mathcal{M})$, which contradicts $u$ being an $n$-chain. It remains $g = t_2$ implying $u \in \mathcal{L}(\mathcal{M}')$, and the second condition $s' = \cup_{t_2=t''t'} \mathcal{M}_{t'}(q_1)$ is also satisfied. The induction principle asserts that every $n$-chain belongs to $\mathcal{L}(\mathcal{M}')$.

For the opposite inclusion, suppose $u \in \mathcal{L}(\mathcal{M}')$. We want to prove that $u$ is a chain, i.e an $n$-chain for some $n \geq 1$. Let $u = t_1t_2 \cdots t_n$ be the complete factorization of $u$ in the sense that $\{t_1t_2 \cdots t_m | 1 \leq m \leq n\}$ is the set of all left subwords of $u$ which belong to $\mathcal{L}(\mathcal{M}')$. We use induction on $n$. It is easy to realize $u$ is a 1-chain for $n = 1$, so we make the following induction assumption for $m < n$:

- The word $t_1t_2 \cdots t_m$ is an $m$-chain with tail $t_m$.
- If $\mathcal{M}'_{t_1t_2 \cdots t_m}(q'_1) = q'_{(r,s)}$ then $s = \cup_{t_m=t''t'} \mathcal{M}_{t'}(q_1)$.

Also here we note that for $m = 1$ the first statement is not valid, since $t_1$ is not its own tail. As we will see this will not cause any problems with the proof, since $\mathcal{L}(\mathcal{M})$ is reduced and thus $t_1$ does not contain a proper subword in $\mathcal{L}(\mathcal{M})$.

Let $\mathcal{M}'_{t_1 t_2 \cdots t_{n-1}}(q'_1) = q'_{(r,s)}$. Then $\mathcal{M}'_{t_1 t_2 \cdots t_n}(q'_1) = \mathcal{M}'_{t_n}(q'_{(r,s)}) = q'_{(r',s')}$, where

$$\begin{cases} r' = \cup_{t_{n-1}=t''t', t'' \neq t_{n-1}} \mathcal{M}_{t't_n}(q_1) \\ s' = \cup_{t_n = t''t'} \mathcal{M}_{t'}(q_1) \end{cases}$$

Now $r' \cap F \neq \emptyset$, so for at least one right subword $t'$ of $t_{n-1}$ we have $t't_n \in \mathcal{L}(\mathcal{M})$. Assume there is another word $g \in \mathcal{L}(\mathcal{M})$ dividing $t_{n-1}t_n$. The case $g|t_{n-1}$ is impossible since $t_{n-1}$ is a tail of some chain, and so is $g|t_n$ since $t't_n \in \mathcal{L}(\mathcal{M})$, which is a reduced language (notice that $t'$ is nonempty). It remains the case where $g$ intersects both $t_{n-1}$ and $t_n$. By the construction of the factorization $t_1 t_2 \cdots t_n$, $g$ must be another right subword of $t_{n-1}t_n$. But then $g = t't_n$ since $\mathcal{L}(\mathcal{M})$ is reduced. Thus $t_1 t_2 \cdots t_n$ is an $n$-chain with tail $t_n$ and in addition $s' = \cup_{t_n = t''t'} \mathcal{M}_{t'}(q_1)$, satisfying the second part of the induction assumption. By induction $u$ is an $n$-chain and we are done. □

**Remark** Let $\mathcal{M}$ be a DFSA for the $n$-chains of some reduced language. The proof of proposition 3.2d reveals the following pleasant property of $\mathcal{L}(\mathcal{M})$. Let $u \in \mathcal{L}(\mathcal{M})$ and let further $u = t_1 t_2 \ldots t_n$ be the complete factorization of $u$ as described before. Then $u$ is an $n$-chain. In this way we can keep track of the index of the chain by counting the number of terminal states reached during the run of $\mathcal{M}$.

### 3.3 Aspects on time complexity

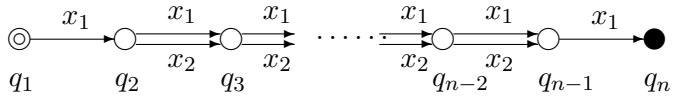
As noted before the complexity for the algorithm in proposition 3.1 as well as for the variants used in proposition 3.2 is exponential in the input size. But in many cases we will have much lower average complexity, taking advantage of special properties of the given input. All of the algorithms in 3.2 can be improved considerably considering the fact that all states not reachable from the initial state safely can be

removed. This can be accomplished by constructing the new automaton stepwise, using some kind of search of the corresponding graph, e.g a depth-first search starting from the initial state. Any state not reachable from the initial state will thus never be taken account of. As concerns the automaton for the complement of normal words in b), we might also use the fact that any transition from a terminal state must map to another terminal state. Thus it is possible to backtrack as soon as a terminal state is reached. Another way to keep complexity down is to find equivalent automata of minimum size, i.e with a minimum number of states. A polynomial algorithm for this can be found in [3].

Although the efficiency can be improved considerably in many cases the following proposition prohibits general success.

**Proposition 3.3** *Given a DFSA* $\mathcal{M} = \{Q, X, q_1, F\}$ *with* $|Q| = n$ *the time complexity for computing a DFSA for the language of normal words with respect to* $\mathcal{L}(\mathcal{M})$ *is* $\Theta(2^n)$.

**Proof** Let $\mathcal{M}$ be defined by the following graph.




We compute a DFSA $\mathcal{M}' = \{Q', X, q_1', F'\}$ which recognizes the set of normal words using the algorithm in proposition 3.2. We maintain the notions introduced in proposition 3.1. Let $q_{i_1}, q_{i_2}, \dots q_{i_k}$ $(1 \leq k \leq n-1)$ be a sequence of states in $Q$ such that $1 = i_1 < i_2 < \cdots < i_k < n$. Consider the word

$$u = x_1 x_2^{i_k - i_{k-1} - 1} x_1 x_2^{i_{k-1} - i_{k-2} - 1} \cdots x_1 x_2^{i_2 - i_1 - 1}$$

It is clear that $\mathcal{M}'_u(q_1') = q'_{\{q_{i_1}, q_{i_2}, \dots q_{i_k}\}}$ which means all $2^{n-2}$ terminal states $F' = \{q_r' \in Q' | q_1 \in r,\ q_n \notin r\}$ are reachable from $q_1'$.

Now suppose $\mathcal{M}'' = \{Q'', X, q_1'', F''\}$ is another DFSA recognizing the same language, and $q'' \in Q''$ a state reachable from $q_1''$. Moreover,

let $u_1, u_2 \in X^*$ be normal words satisfying $\mathcal{M}''_{u_i}(q''_1) = q''$ $(1 \leq i \leq 2)$. For $\mathcal{M}'$ we have

$$\mathcal{M}'_{u_i}(q'_1) = q'_{r_i},\ r_i = \{q_{l_{i1}}, q_{l_{i2}}, \ldots, q_{l_{ik_i}}\}$$

$$\text{with } 1 = l_{i1} < l_{i2} < \cdots < l_{ik_i} < n,\ \ 1 \leq i \leq 2,\ \ 1 \leq k_i \leq n-1$$

We want to prove $\mathcal{M}'_{u_1}(q'_1) = \mathcal{M}'_{u_2}(q'_1)$. Suppose not. By symmetry we can safely assume $r_1 \not\subseteq r_2$, so for some $j \neq 1$, $q_{l_{1j}}$ belongs to $r_1$ but not to $r_2$. Consider the word $v = x_2^{n-l_{1j}-1}x_1$. It follows that

$$\begin{cases} \mathcal{M}'_{u_1v}(q'_1) = q'_{\{q_1, q_{l_{11}+1}, q_{l_{12}+(n-l_{1j})}, \ldots, q_{l_{1(j-1)}+(n-l_{1j})}, q_n\}} \notin F' \\ \mathcal{M}'_{u_2v}(q'_1) = q'_{\{q_1, q_{l_{21}+1}, q_{l_{22}+(n-l_{1j})}, \ldots, q_{l_{2m}+(n-l_{1j})}\}} \in F' \end{cases}$$

where $l_{2m}$ is the largest integer strictly less than $l_{1j}$. But $\mathcal{M}''_{u_1v}(q''_1) = \mathcal{M}''_{u_2v}(q''_1)$ which contradicts $\mathcal{L}(\mathcal{M}') = \mathcal{L}(\mathcal{M}'')$. Thus $\mathcal{M}'_{u_1}(q'_1) = \mathcal{M}'_{u_2}(q'_1)$. We have proved that for any normal words $u_1, u_2 \in X^*$

$$\mathcal{M}'_{u_1}(q'_1) \neq \mathcal{M}'_{u_2}(q'_1) \Rightarrow \mathcal{M}''_{u_1}(q''_1) \neq \mathcal{M}''_{u_2}(q''_1)$$

This means $|F''| \geq |F'| = 2^{n-2}$ so any DFSA recognizing the set of normal words needs at least $2^{n-2}$ states. □

## 4 Hilbert series and Poincaré series

In this section we will provide general methods for computing Hilbert series and Poincaré series by using the automata produced in the previous section. Most of the major ideas as regards the use of automata to compute Hilbert series are already known, but presented in a somewhat more specialized context (see [8]). In addition to presenting a unified theory, we will provide examples how to compute Hilbert series for subalgebras generated by a rational language, as well as discuss various implementational aspects.

### 4.1 Hilbert series computation for rational languages

Assuming a somewhat more general approach, we extend the definition of Hilbert series to an arbitrary language $\mathcal{L}$.

**Definition 4.1** *Let $\mathcal{L}$ be a language. We define the Hilbert series of $\mathcal{L}$ as the formal power series*

$$H_{\mathcal{L}}(t) = \sum_{u \in \mathcal{L}} t^{\deg u}$$

As mentioned earlier every DFSA $\mathcal{M} = \{Q, X, q_1, F\}$ can be represented as a graph $\mathcal{G}(\mathcal{M})$. This can be used to effectively compute the Hilbert series of $\mathcal{L}(\mathcal{M})$. Let $Q = \{q_1, q_2, \ldots q_n\}$ and let $D$ be the $n \times n$-matrix defined by

$$D_{ij} = \sum_{x \in X,\, \mathcal{M}_x(q_i) = q_j} t^{\deg x}$$

We say that $D$ is the matrix associated with $\mathcal{G}(\mathcal{M})$. We also introduce the $n$-sized vector

$$e = (\phi(1), \phi(2), \ldots \phi(n)),\ \phi(i) = \begin{cases} 1 & \text{if } q_i \in F \\ 0 & \text{if } q_i \notin F \end{cases}$$

The following proposition can be found in [8] in a slightly different form.

**Proposition 4.1** *Let $\mathcal{M}$ be a DFSA recognizing a language $\mathcal{L}$, and $\mathcal{G}(\mathcal{M})$ the corresponding graph. Then*

$$H_{\mathcal{L}}(t) = (I - D)_1^{-1} e^T$$

*where $I$ is the unit matrix and the lower index '1' denotes the first row of the matrix. The upper index 'T' simply indicates $e$ being the transposed vector.*

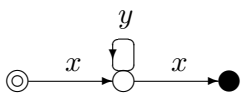
**Proof** It is easy to see that for any $n$, $(D^n)_1 e^T$ equals the Hilbert series for all words in $\mathcal{L}(\mathcal{M})$ of length $n$. Thus

$$H_{\mathcal{L}}(t) = (\sum_{n \geq 0} (D^n)_1) e^T = (\sum_{n \geq 0} D^n)_1 e^T = (I - D)_1^{-1} e^T \quad \square$$

## 4.2 Finitely presented algebras

Let $A = \oplus_{n\geq 0} A_n$ be a graded algebra and $H_A(t) = \sum_{n\geq 0} \dim(A_n)t^n$ its Hilbert series. If $A$ is a finitely presented algebra, i.e $A = K\langle X\rangle/I$ for some finitely generated homogeneous ideal $I$, one could expect the Hilbert series to be rational. That is surprisingly not the case. For a counterexample see [5]. For algebras with a rational language of normal words rationality is guaranteed. Hilbert series computations for this class of algebras are done in [7]. However, a complete algorithmic procedure is only described for algebras with a finite Gröbner basis. Obviously a Gröbner basis with a rational set of leading words also results in a rational set of normal words, and experimental results indicate such bases are frequently occurring. We provide an example:

**Example 4.1** *The algebra $A = K\langle x, y|x^2 = xy\rangle$ has, using deglex order with $x > y$, a Gröbner basis $G = \{xy^n x = xy^{n+1}|n \in \mathbb{N}\}$. The set $\hat{G}$ of leading words is rational and can be recognized by an appropriate DFSA:*




**Remark** In general infinite Gröbner bases create serious problems with computer calculations. One possible way to get around this is discussed in [8]. The main idea is to make the computer guess a 'rational' Gröbner basis based on the initial Gröbner basis computations, and then prove its correctness using the diamond lemma [2]. A computer program for this 'guessing'-procedure has been implemented by the same author. Also this approach is based on the fact that many algebras seem to admit a rational structure of their minimal Gröbner basis. It would be interesting to be able to in some sense predict whether a given finitely presented algebra admits a 'rational' Gröbner basis or not. Obviously this not a general property.

Let $I$ be a homogeneous two-sided ideal in $K\langle X\rangle$ and let $N$ be the set of normal words modulo $I$. If $KN$ denotes the $K$-linear span of $N$, it is easy to prove that the following direct sum of vector spaces hold (see e.g [9]):

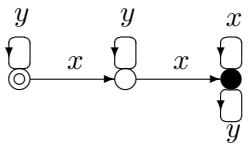
$$K\langle X =\rangle = KN \oplus I$$

If we introduce a new operation on $KN$ by letting $s * t = \overline{st}$, where the bar denotes the projection on $KN$, $KN$ is isomorphic to the factor algebra $A = K\langle X\rangle/I$. The advantage with this identification is the possibility to compute the Hilbert series of $A$ directly from $N$. In fact we get

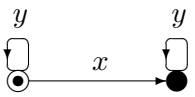
$$H_A(t) = H_N(t)$$

Thus, according to proposition 4.1, we can compute the Hilbert series of $A$ once we have a DFSA for its set of normal words.

**Example 4.2** *We continue with example 4.1. Using the algorithm in 3.2a, the initial ideal of* $A = K\langle x, y|x^2 = xy\rangle$ *can be represented by*




*To obtain a deterministic automaton for the set of normal words, we interchange terminal and nonterminal states (notice that all nonterminal states in the new automaton safely can be removed)*




*yielding an associated matrix* $D = \begin{pmatrix} t & t \\ 0 & t \end{pmatrix}$. *According to proposition 4.1 we get*

$$H_A(t) = H_N(t) = (I - D)_1^{-1} e^T = (\frac{1}{1-t}, \frac{t}{(1-t)^2}) \begin{pmatrix} 1 \\ 1 \end{pmatrix} =$$

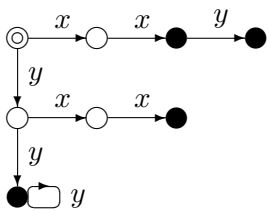
$$= \frac{1}{(1-t)^2}$$

By means of the techniques developed in this paper we can also compute Hilbert series for monomial subalgebras of $K\langle X\rangle$ generated by a rational set of monomials. For subalgebras of $K\langle X\rangle$ the analogue of Gröbner bases is called SAGBI-bases (**S**ubalgebra **A**nalogue of **G**röbner **B**ases for **I**deals). More about this concept can be found in [4].

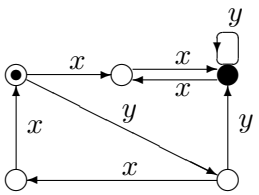
**Corollary 4.1** *For any subalgebra of $K\langle X\rangle$ with a rational SAGBI-basis, the Hilbert series is rational.*

We illustrate with an example:

**Example 4.3** *Let $A$ be the subalgebra generated by $\mathcal{L} = \{x^2, x^2y, yx^2, y^{2+n} | n \in \mathbb{N}\}$.*




*¿From algorithm 3.2b we get that $\mathcal{L}^*$ is recognized by (after minimization)*




*We can now compute the Hilbert series with the aid of proposition 4.1.*

$$H_A(t) = H_{\mathcal{L}^*}(t) = (I - D)_1^{-1} e^T = \frac{t^2 - t + 1}{t^5 + t^4 - t^3 - t^2 - t + 1}$$

### 4.3 Poincaré series

The Poincaré series for algebras is defined in the following way:

**Definition 4.2 (Poincaré series)** *The* Poincaré series *of a graded algebra $A$ is determined by*

$$P_A(t) = \sum_{n \geq 0} \dim_K(Tor_n^A(K,K))t^n$$

In order to get a good upper bound on the Poincaré series, we are also interested in the following more general series:

**Definition 4.3 (Double Poincaré series)** *The* double Poincaré series *of a graded algebra $A$ is the following series in two variables:*

$$P_A(s,t) = \sum_{i,n \geq 0} \dim_K(Tor_{in}^A(K,K))s^i t^n$$

The Poincaré series reflects the homological properties of an algebra $A$ and carries a great amount of information. By considering a free resolution of $A$, the double Poincaré series can be computed explicitly. In particular, so called minimal resolutions are very important for this matter. In [1] Anick constructed a resolution with modules generated by the $n$-chains of $A$. The following is true for monomial algebras:

**Proposition 4.2** *Let $A = K\langle X|F\rangle$ be a monomial algebra. Then*

$$P_A(s,t) = 1 + H_{C_0}(t)s + H_{C_1}(t)s^2 + \cdots$$

*where $C_i$ denotes $C_i(F_R)$.*

Note that $F$ here can be any set of monomials, not necessarily finite or rational. For a more detailed exposition on Anick's resolution, see also [9]. The proof of the following remarkable formula can also be found there.

**Proposition 4.3** *Let $A = K\langle X\rangle/I$ be a finitely generated homogeneous algebra, and $G_R$ the reduced Gröbner basis for $I$. Then*

$$H_A(t) = \frac{1}{1 - \sum_{i \geq 0}(H_{C_{2i+1}(\hat{G}_R)}(t) - H_{C_{2i}(\hat{G}_R)}(t))}$$

As we can see, the concept of $n$-chains can be used to compute both Hilbert series and double Poincaré series. Taking advantage of the remark following proposition 3.2 we can prove the following proposition:

**Proposition 4.4** *Let $A = K\langle X\rangle/I$ be a finitely generated homogeneous algebra and $G_R$ the reduced Gröbner basis for $I$. If $\mathcal{M} = \{Q, X, q_1, F\}$ is a DFSA for the language of $n$-chains of $\hat{G}_R$, then we can construct an associated matrix $D$ and a vector $e$ such that*

*a)* $H_A(t) = 1/(1 - H_X(t) - (I - D)_1^{-1}e^T)$

*b)* $P_{\tilde{A}}(s,t) = 1 + H_X(t)s + s(I - D)_1^{-1}e^T$

*where $\tilde{A} = K\langle X|\hat{G}_R\rangle$, i.e the corresponding monomial algebra.*

**Proof**
a) Assume $Q = \{q_1, q_2, \ldots, q_n\}$. We define the $n \times n$-matrix $D$ by

$$D_{ij} = \sum_{x\in X, \mathcal{M}_x(q_i)=q_j} (-1)^{\phi(j)} t^{\deg x}$$

and let $e = (\phi(1), \phi(2), \ldots, \phi(n))$. Recall that

$$\phi(i) = \begin{cases} 1 & \text{if } q_i \in F \\ 0 & \text{if } q_i \notin F \end{cases}$$

Just like in the proof of proposition 4.1 we notice that $(D^n)_1 e^T$ equals the Hilbert series of $\mathcal{L}(\mathcal{M})$, with one major exception. Since any $n$-chain ($n \geq 1$) corresponds to a traversion of $\mathcal{G}(\mathcal{M})$ passing $n$ terminal states, and thus changing sign the same number of times, we obtain instead

$$(I - D)_1^{-1}e^T = \sum_{i\geq 1} (-1)^i H_{C_i(\hat{G}_R)}(t)$$

Combining this and proposition 4.3 yields

$$H_A(t) = \frac{1}{1 - H_X(t) - (I - D)_1 e^T}$$

which was desired. Note that $H_{C_0(\mathcal{L})}(t) = H_X(t)$.

b) We define $D$ by

$$D_{ij} = \sum_{x \in X, \mathcal{M}_x(q_i)=q_j} s^{\phi(j)} t^{\deg x}$$

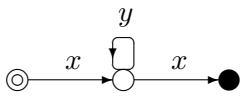
and let $e$ be as in a). Using proposition 4.2, a similar argument as in a) implies

$$P_{\tilde{A}}(s,t) = 1 + H_X(t)s + s(I-D)_1^{-1}e^T$$

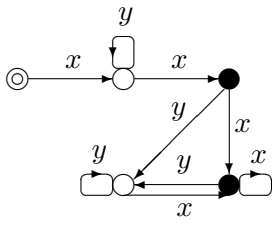
and we are done. □

Proposition 3.2 and 4.4 now provide us with a new algorithmic procedure to compute monomial Poincaré series for any given automaton algebra.

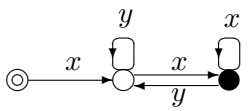
**Example 4.4** *We return to the algebra $A = K\langle x, y | x^2 = xy\rangle$. As was seen in example 4.2 $A$ has the reduced Gröbner basis $G = \{xy^n x = xy^{n+1} | n \in \mathbb{N}\}$ and thus a set of leading words $\hat{G} = \{xy^n x | n \in \mathbb{N}\}$.*




*Using the algorithm in proposition 3.2d we obtain the following DFSA for the language of n-chains:*




*which after minimization is equivalent to*




*By means of proposition 4.4 we obtain*

$$H_A(t) = 1/(1 - H_X(t) - (I - D)_1^{-1} e^T) = 1/(1 - 2t - (-t^2)) =$$

$$= 1/(1-t)^2$$

$$P_{\bar{A}}(s,t) = 1 + H_X(t)s + s(I - D)_1^{-1} e^T = 1 + 2st + \frac{s^2 t^2}{1 - (st + t)}$$

# References


[1] Anick, D. (1986): *On the homology of associative algebras*, Trans. Am. Math. Soc. *296*, No. 2, pp.641-659.

[2] Bergman, G. (1978): *The diamond lemma for ring theory*, Adv. Math. *29*, No. 2, pp.178-218.

[3] Drobot, V. (1989): *Formal languages and automata theory*, Computer Science Press.

[4] Nordbeck, P. (1999): *Canonical bases for subalgebras of factor algebras*, Comput. Sci. J. Mold. *7*, pp.63-77.

[5] Shearer, J.B. (1980): *A graded algebra with a non-rational Hilbert series*, J. Algebra *62*, No. 1, pp.228-231.

[6] Tsejtin, G.S. (1958): *Associative computations with unsolvable equivalency problem*, Tr. Mat. Inst. Steklova *52*, pp.172-189.

[7] Ufnarovski, V.A. (1991): *On the use of graphs for computing a basis, growth and Hilbert series of associative algebras*, Math. USSR Sbornik *68*, No. 2, pp.417-428.

[8] Ufnarovski, V.A. (1993): *Calculations of growth and Hilbert series by computer*, Lect. Notes Pure Appl. Math. *151*, pp.247-256.

[9] Ufnarovski, V.A. (1995): *Combinatorial and asymptotic methods of algebra*, Algebra-VI (A.I. Kostrikin and I.R. Shafarevich, Eds), Encyclopedia of Mathematical Sciences, Vol. 57, Springer, pp.5-196.

Jonas Månsson,
Lund University,
Department of Mathematics
Sölvegatan, 18,
Box 118, S–22100,
Lund, Sweden
e-mail: *jonasm@maths.lth.se*

Received April 3, 2000