# The automata that define representations of monomial algebras

Sarah Rees

8th June 2007

**Abstract**

It is well known that the sets of strings that define all representations of string algebras and many representations of other quotients of path algebras form a regular set, and hence are defined by finite state automata. This short article aims to explain this connection between representation theory and automata theory in elementary terms; no technical background in either representation theory or automata theory is assumed.

The article describes the structure of the set of strings of a monomial algebra as a locally testable and hence regular set, and describes explicitly the construction of the automaton, illustrating the construction with an elementary example. Hence it explains how the sets of strings and bands of a monomial algebra correspond to the sets of paths and closed (non-powered) circuits in a finite graph, and how the growth rate of the set of bands is immediately visible from that graph.

## 1 Introduction

The purpose of this short article is to demonstrate an automata-theoretic perspective on the representations of certain algebras. The algebras in questions are path algebras (that is algebras of linear combinations of paths in a finite directed graph $Q$) and some of their quotients. A representation of such an algebra $A$ is simply an algebra homomorphism into the algebra of endomorphisms of a vector space, which can then be viewed as a module for $A$.

An article of Butler and Ringel [1] studied particular quotients of path algebras, termed string algebras, and showed that any indecomposable representation of a string algebra can be described in terms of an undirected path within the universal cover $\tilde{Q}$ of the directed graph $Q$, which satisfies certain constraints (basically its exclusion of certain subpaths), and which is known as a string. Distinct strings yield non-isomorphic representations. Under certain conditions, a string is also a band, in which case a parametrised family of representations can also be defined from it. For a string algebra, every indecomposable module is either a string module or a band module.

Since the graphs $Q$ from which the algebras being considered are defined are finite graphs, the paths in $\tilde{Q}$ from which the representations are defined can (with the exception of the paths of length 0, which we shall ignore in this article; they yield only finitely many representations) be described as sequences over a finite set (the set of edges of $Q$ and their reverses). It is fairly well known that the set of such sequences which define the strings for a monomial algebra is a regular set. This means that it is the (usually infinite) set of sequences recognised by a finite state automaton [3], that is, a theoretical computer with bounded memory; such an automaton can be interpreted simply as a finite directed edge-labelled graph, and its accepted sequences as the sequences labelling directed paths through that graph from the 'start' vertex to an 'accepting' vertex. The regular set of strings also turns out to have a particularly elementary structure, that is, to be locally testable [4].

We shall show how to construct an automaton recognising the strings of length greater than 0 for a monomial algebra $A$ defined from a given finite directed graph $Q$. We describe the automaton graphically. We shall also prove that a string is a band if and only if it labels the edges of a closed directed circuit in the (automaton) graph $G$.

Hence we find the strings and bands of a monomial algebra as unconstrained paths and circuits of a finite graph $G$, rather than as certain constrained subgraphs of the (probably infinite) graph $\tilde{Q}$. Using the finite graph $G$, certain features such as the growth rate of the bands (and hence of the parametrised families of representations) then become immediately visible.

As an illustration we describe the automaton graph for an elementary example, a string algebra for which the growth rate of bands is clearly seen to be exponential.

This article arose from a visit to Paderborn in 2004,5. I am grateful to Henning Krause for suggesting the topic, to him and others in Paderborn, especially Markus Perling and visitor Ed Green, for very helpful discussions. I should also like to thank the Universität Paderborn and the Deutsche Akademische Austauschdienst (DAAD) for their hospitality and support.

I have included basic definitions both from the theory of algebras and their representations and from automata theory, in order to make this accessible to a wider audience. Note that throughout the article, maps are composed from left to right and sit on the right of their arguments. Although this may seem non-standard usage for representations of algebras, it is consistent with standard usage in combinatorics and automata theory, where paths in graphs and sequences input to automata are in general read from left to right, and hence seems appropriate in the context of the article.

## 2 Background on algebras and their representations

We suppose that $Q$ is a finite, directed graph, with vertex set $Q_0$ and set $Q_1$ of directed edges. Loops and multiple edges are permitted in $Q$.

We denote the source of a directed edge $a$ of $Q$ by $s(a)$ and its terminus by $t(a)$. A sequence of directed edges $a_1 \dots a_m$ defines a directed path $p$ in $Q$ if for each $i = 1, \dots m-1$, $t(a_i) = s(a_{i+1})$. Then $p$ has source $s(p) = s(a_1)$, terminus $t(p) = t(a_m)$ and length $m$. The concatenation $pq$ of $p$ and $q$ is defined in the obvious way provided that $t(p) = s(q)$. Otherwise we say that $pq = 0$.

It is convenient to define the reverse $a^{-1}$ of an edge $a$, setting its source to be the terminus of $a$ and its terminus to be the source of $a$, and to define the reverse $p^{-1}$ of a path $p = a_1 \dots a_m$ to be the concatenation of reverse edges $a_m^{-1} \dots a_1^{-1}$, with source $t(p)$ and terminus $s(p)$. It is also convenient to consider the edge $a$ to be the reverse of the reverse edge $a^{-1}$.

Where $Q$ is a directed graph and $k$ a field (assumed algebraically closed) the path algebra $kQ$ is defined to be the $k$-algebra spanned by the directed paths in $Q$, with multiplication of paths defined as above. $Q$ is then called the Gabriel quiver of $kQ$ (and of its quotients). Its directed edges are usually called arrows (of $Q$ and of $kQ$ and its quotients).

Suppose that $P$ is a set of paths, each of length at least 2. Then the monomial algebra $kQ/\langle P\rangle$ is defined to be the quotient of $kQ$ defined by setting every path in $P$ to be zero. The path algebra itself is a monomial algebra, formed by taking $P$ to be the empty set.

The monomial algebra $kQ/\langle P\rangle$ is finite dimensional provided that, for some constant $N$, any directed path in $Q$ of length $N$ contains a subpath in $P$. If in addition, (a) every vertex in $Q$ is the source of at most two arrows and also the terminus of at most two arrows and (b) given any arrow $b$ there is at most one arrow $a$ with $t(a) = s(b)$ and $ab$ outside $P$, and also at most one arrow $c$ with $s(c) = t(b)$ and $bc$ outside $P$, then $kQ/\langle P\rangle$ is called a string algebra. In that case every indecomposable module for $kQ/\langle P\rangle$ is either a string module or a band module (as defined below). Otherwise there will also be other modules.

### 2.1 String and band modules

We find modules for (and hence representations of) a monomial algebra $A = kQ/\langle P\rangle$ as direct sums $V = \oplus X_i$ of $k$-vector spaces, indexed by the vertices of $Q$, such that each arrow $a$ of $A$ acts as a linear map from $X_{s(a)}$ to $X_{t(a)}$.

The modules under consideration in this article are most easily defined in relation to the universal cover of $Q$.

A directed graph $\hat{Q}$ is said to cover $Q$ if a graph morphism $\theta : \hat{Q} \to Q$ can be defined whose restriction to the set of edges through a vertex is a bijection; $\theta$ is then called a covering morphism. The universal cover $\tilde{Q}$, which covers all connected covers of $Q$, is the unique directed tree locally isomorphic to $Q$. It is natural to label vertices and edges of $\tilde{Q}$ according to the vertices and edges they map onto in $Q$, that is by the sets $Q_0$ and $Q_1$. We can identify the set $\tilde{P}$ of directed paths in $\tilde{Q}$ which map under the covering morphism to paths in $P$.

Now we define a *string* of length $m > 0$ to be a connected subgraph of $\tilde{Q}$ on $m$ edges, in which every vertex has degree 1 or 2, and which does not contain any path in $\tilde{P}$ as a subgraph. We consider two paths which are related by a deck transformation of $\tilde{Q}$ (that is, an automorphism $\alpha$ of $\tilde{Q}$ which permutes the fibres of the covering map) to be equivalent.

Equivalently a string is an undirected path within $\tilde{Q}$ which contains no backtracking (that is no edge followed by its reverse) and no element of $\tilde{P}$ or reverse of such an element as a subpath; that is, a string $s$ is a concatenation $u_1 \dots u_k$ such that either $u_1, u_3, \dots u_{2i-1}, \dots$ are paths within $\tilde{Q}$ and $u_2, u_4, \dots u_{2i}, \dots$ are the reverses of paths, or vice versa, and such that for each $j$, the terminus of $u_j$ is equal to the source of $u_{j+1}$, and no path in $\tilde{P}$ or reverse of a path in $\tilde{P}$ is a subpath of any $u_i$. Using this point of view we see that a string can be represented as a sequence of elements from the set $Q_1 \cup Q_1^{-1}$. It fact it can be represented by two such sequences, because an undirected path and its reverse in $\tilde{Q}$ correspond to the same subgraph and hence are equivalent as strings.

Note that we have not defined strings of length 0, because we shall not consider them in this paper. There are finitely many such.

We define a string $s$ to be a *band* if its image under the covering morphism is a closed path, but not a power of a closed path, and in addition every power of $s$ is also a string. (Note that since $s$ maps to a closed path in $Q$, any power $s^k$ of $s$ is naturally defined as the extension of $s$ which maps onto the $k$-th power of that closed path in $Q$.)

For any monomial algebra $A$, each string defines an indecomposable representation of $A$, each band defines a parametrised family of representations, and inequivalent strings and bands or distinct parameter sets define inequivalent representations. For a string algebra, these representations are the only indecomposable representations of $A$, but for a more general monomial algebra there are also other representations (in particular, the string and band representations can be easily generalised to representations defined in terms of more general subgraphs of $\tilde{Q}$ than the paths which define strings [2]).

Precisely, given a string $s = x_1 \dots x_n$ of length $n \geq 1$, where each $x_i$ is either an edge or a reverse of an edge in $Q$, we define an $n+1$-dimensional indecomposable representation for $A$ with basis $e_1, \dots e_{n+1}$, by defining, for each arrow $a$ of $Q$, $e_i^a = e_{i+1}$ if $x_i = a$, $e_i^a = e_{i-1}$ if $x_{i-1} = a^{-1}$, and otherwise $e_i^a = 0$.

When $s$ is also a band, for any non-zero $\nu$ in $k$, and any integer $m \geq 1$, we define a further parametrised representation of dimension $mn$ as follows. For each $i = 1, \dots n$, we define an $m$-dimensional vector space $E_i$ over $k$, selecting a basis of this. We define $I_m$ to be the $m \times m$ identity matrix, and $J_{\nu m}$ to be the $m \times m$ Jordan block matrix with eigenvalue $\nu$ (that is the $m \times m$ matrix with each entry on the leading diagonal equal to $\nu$, each entry just above the leading diagonal equal to 1, and all other entries 0). Then for each arrow $a$ of $Q$, considering all indices modulo $n$, we define $a$ to map $E_i$ to $E_{i+1}$ if $x_i = a$, to map $E_i$ to $E_{i-1}$ if $x_{i-1} = a^{-1}$ and to act trivially on $E_i$ otherwise. If $a$ acts non-trivially on $E_i$, then its action is usually defined, with respect to the selected bases, by the identity matrix $I_m$, except that if $x_1 = a$, then $a$ acts as $J_{m\nu}$ mapping $E_1$ to $E_2$, and if $x_1 = a^{-1}$, then $a$ acts as $J_{m\nu}$ mapping $E_2$ to $E_1$.

## 3 A little automata theory

In this section we introduce just the small amount of automata theory which we need in the rest of the article. More details can be found in [3, 4]

We define a (deterministic) finite state automaton $G$ (we call it $G$ because we shall think of it as a finite graph) to be a device with a finite number of states from a set $V$, one of which, $v_0$, is called the start state, and some of which are accept states, some reject states. The automaton takes as its input sequences $x_1 \dots x_n$ of elements from a finite set $X$, reading the symbols within the string one at a time, starting with $x_1$. It starts reading each sequence in the state $v_0$, and after reading each symbol chooses the next state according to the state it is currently in and the symbol it has just read. It accepts a sequence if it is in an accept state when it reaches the string's end. We use the symbol $\epsilon$ to denote the empty sequence (which is valid input, accepted if and inly if $v_0$ is an accept state).

An automaton as described above can clearly be modelled by a finite graph (possibly with loops and multiple edges), whose vertices are labelled by the states in $V$, and whose directed edges are labelled by the elements of $X$, and determine the transitions between states. A string is then accepted if it labels a directed path from the vertex $v_0$ to a vertex representing an accept state. In a situation where any prefix of an accepted sequence is also accepted, we can achieve a more compact representation by deleting all non-accept states, and all edges into them; a sequence is then accepted by the automaton if it labels a directed path which starts at $v_0$. We shall use this more compact, graphical representation in this article.

Many automata can be defined accepting any given set of sequences, but there is a unique minimal automaton. The minimal automaton $G_{min}$ accepting the same set of sequences as $G$ is formed from $G$ by repeatedly merging any two states $v, v'$ with the same 'future', that is for which the set of sequences labelling a route from $v$ to an accept state is equal to the set of sequences labelling a route from $v'$ to an accept state.

A set of sequences is called regular if and only if it is the set of sequences accepted by a finite state automaton. Various subclasses of the class of regular sets are of interest, and well understood. One such is the class of locally testable sets; a set of sequences is $k$-testable if the membership of a sequence $x_1 \dots x_n$ in that set is determined by the character of its subsequences $x_i \dots x_{i+k-1}$ of length $k$ (we shall call such subsequences *subwords*). The class of locally testable (that is $k$-testable for some $k$) sets lies within the class of star-free sets; a regular set is star-free if there is no loop within the minimal automaton accepting it which is labelled by a proper power $w^r$, with $r > 1$. It will soon be clear that the regular sets of strings are locally testable, and hence star-free.

## 4 Describing the automata

From now on we assume that the algebra $kQ/\langle P \rangle$ is finite dimensional, and hence in effect that $P$ is finite (we can ignore any elements of $P$ of length

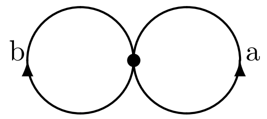


Figure 1: the Gabriel quiver

more than the given bound $N$). In that case, a sequence $x_1 \dots x_n$ of elements of $Q_1 \cup Q_1^{-1}$ is a string provided that (a) for each $i$, $t(x_i) = s(x_{i+1})$, (b) for each $i$, $x_{i+1}$ is not the reverse of $x_i$, (c) no subword of length $N$ or less is in $P$. Hence the set of strings is $k$-testable, where $k$ is the maximum of 2 and $N$.

Hence it should be clear that the following construction gives an automaton $G$ recognising the strings of $kA/\langle P \rangle$.

We use the compact graphical representation described above (so in particular all the states we represent are accept states). We label the start state by $v_0$. Each other state is labelled by a pair $(x, w)$, where $x \in Q_1 \cup Q_1^{-1}$ and $w$ is a (possibly empty) sequence of length less than $N$ over either $Q_1$ (if $x \in Q_1$) or $Q_1^{-1}$ (if $x \in Q_1^{-1}$). For each $y \in Q_1 \cup Q_1^{-1}$ we have an edge labelled $y$ from $s_0$ to either $(y, y)$ if there is a path in $P \cup P^{-1}$ which starts with $y$ or $(y, \epsilon)$ otherwise. More generally, from $(x, w)$, we have no edge labelled by $y$ if the terminus of $x$ is not the source of $y$, or if $y$ is the reverse of $x$, or if $wy$ is a path in $P \cup P^{-1}$. Otherwise if one of $x$ and $y$ is in $Q_1$ and the other in $Q_1^{-1}$, we have an edge labelled $y$ from $(x, w)$ to $(y, y)$, and if either both are in $Q_1$ or both in $Q_1^{-1}$ we have an edge labelled $y$ from $(x, w)$ to $(y, w')$, where $w'$ is the longest suffix of $wy$ which is a prefix of at least one path in $P \cup P^{-1}$.

The automaton we have just described may not be minimal but can of course be minimised.

As an example, let $Q$ be the quiver shown in figure 1 and let $P$ be the set of paths $\{ab, ba, a^3, b^3\}$. The monomial algebra $kQ/\langle P \rangle$ is a string algebra. The above construction gives the automaton shown in figure 2 which is, in fact, minimal. The start state of the automaton has been marked with an arrow.

# 5 Finding bands within the string automaton

Now let $A = kQ/\langle P \rangle$ be a monomial algebra, and $G$ any finite state automaton recognising its strings. At this stage we do not demand that the

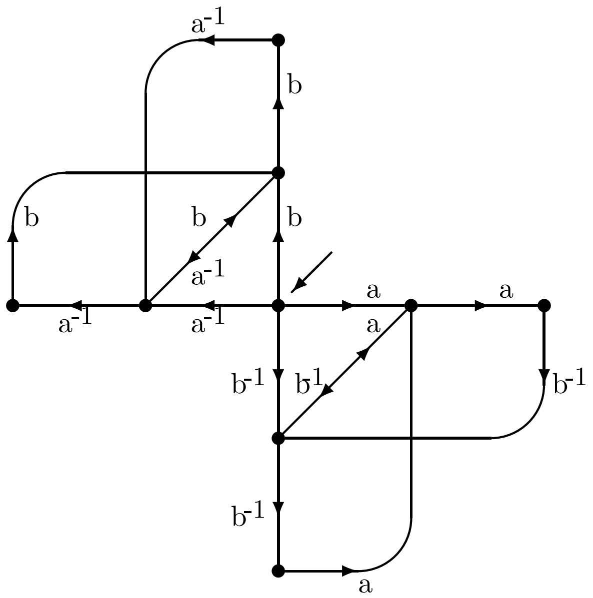


Figure 2: the automaton recognising strings

automaton $G$ be minimal, but we shall assume that it does not contain any inaccessible states (states which are not accessible from the start state $v_0$) or any dead end states (states from which no accept state is accessible). Any inaccessible or dead end states could anyway be pruned from the automaton without changing its operation.

By definition, a sequence over $Q_1 \cup Q_1^{-1}$ is a string if and only if it can be traced out from the start state of the automaton $G$. But since clearly any subsequence of a string is also a string, it is in fact true that any sequence over $Q_1 \cup Q_1^{-1}$ labelling any directed path in $G$ is a string, irrespective of where the path starts.

If $v$ is a state of $G$, either $v = v_0$, or $v$ is an accept state with no directed edges leaving it, or $v$ has edges both entering it and leaving it. In the third case, if $x$ is the label of an edge of $G$ into $v$ and $y$ the label of an edge of $g$ out of $v$, then $xy$ is a string, and so in particular $y$ is not the reverse of $x$, and the terminus of $x$ is equal to the source of $y$. Hence there is a single vertex $v'$ of $Q$ which is the source of all directed edges $e$ of $Q$ for which $e$ labels an edge out of $v$ in $G$ or the reverse of $e$ labels an edge into $v$ in $G$, and which is the terminus of all directed edges $f$ of $Q$ for which $f$ labels an edge of $G$ into $v$ or the reverse of $f$ labels an edge of $G$ out of $v$.

This almost means that we can define a morphism from $G$ seen as a directed graph (with directions on its edges inherited from $Q$ and **not** from its operation as an automaton) into $Q$. In fact it doesn't quite mean that if the start state $v_0$ of $G$ has only edges leaving it and none entering it, in which case two edges leaving $v_0$ may map to edges with distinct vertices of $Q$ as their sources. Similarly if a dead end state $v$ has only edges entering it and no edges leaving it, two edges entering $v$ may map to edges with distinct vertices of $Q$ as their termini. However, we could modify $G$, replacing either $v_0$ or any dead end state by more than one state as necessary, to get a finite graph $G'$, which maps via a graph morphism into $Q$, and whose directed paths are labelled by precisely the set of strings of $A$. Of course if we do this we cannot simultaneously require $G'$ to define a minimal automaton.

From now on, suppose that we have chosen $G$ to be a minimal automaton recognising the strings of $A$.

In this case, any closed circuit anywhere in $G$ must be labelled by a band. For any state $v$ on a closed circuit is accessible and not a dead-end, and so corresponds to a vertex $v'$ of $Q$. Now where $b$ is the label of the circuit in $G$ from $v$ to $v$, $b$ also labels a closed (not necessarily directed) path in $Q$ from $v'$ to $v'$. Since every power of $b$ labels a path in $G$, every power of $b$ is a string. And $b$ cannot be a proper power because the set of all strings is star-free and so no circuits in $G$ can be powered. Hence $b$ itself is a band.

Conversely, suppose that $b$ is a string for $A$ which is also a band. Now trace a high enough power $b^k$ of $b$ out from $v_0$ in $G$. Each $b$ in that power must end at an accessible state of $G$ which maps onto a single vertex of $Q$ (which is the terminus of the path in $Q$ labelled by $b$). Since there are only finitely many states in $G$, two of these states must be the same, and so $G$ contains a circuit which is labelled by a power of $b$. But circuits in $G$ cannot be powered, and hence $b$ labels a circuit in $G$.

It follows from this that

**Proposition 5.1** *A sequence over $Q_1 \cup Q_1^{-1}$ is a band if and only if it labels a circuit in $G$.*

Note that the same band could label more than one circuit. But the number of circuits it labels is bounded by the maximum number of preimages of a single vertex in $Q$ which can be found in $G$. So the set of circuits is comparable with the set of bands.

**Proposition 5.2** *$S$ contains infinitely many bands if and only if $G$ contains a pair of intersecting circuits. In that case, there is a constant $C > 1$ such* *that, for infinitely many values of $n$, the number of bands of length $n$ is greater than $C^n$.*

Proof: If $S$ contains infinitely many bands then there is a band $b$ of length greater than the number of states in $G$. Suppose that $b$ labels a circuit starting at the state $v$ of $G$. Then that circuit must intersect itself, so we can decompose it as $p$, a path in $G$ from $v$ to $w$, followed by $q$, a circuit in $G$ from $w$ to $w$, followed by $r$, a path in $G$ from $w$ to $v$. Then $pr$ and $q$ are circuits within $G$ which intersect at $w$.

Conversely, given a pair $c$, $c'$ of circuits in $G$, with point of intersection $w$ and lengths $m, m'$, take any non-periodic binary string (that is a string of 0's and 1's which cannot be written as a power). Suppose that this string has $\lambda_1$ 0's followed by $\mu_1$ 1's, then $\lambda_2$ 0's etc... and finally $\mu_k$ 1's (we could have $\lambda_1 = 0$ or $\mu_k = 0$, but require the other terms to be non-zero). Then the circuit formed by passing from $w$ $\lambda_1$ times round $c$ and then $\mu_1$ times round $c'$ etc. is a band, and we can make it arbitrarily long by making our binary sequence arbitrarily long. Hence we have infinitely many bands. We can ensure that we form a set of distinct bands by requiring that $\mu_1$ is minimal amongst all the $\mu_i$, and $\lambda_k$ maximal amongst all the $\lambda_i$. Then the number of non-periodic binary strings satisfying those constraints is bounded below by an exponential function of $n = m(\lambda_1 + \ldots + \lambda_k) + m'(\mu_1 + \ldots + \mu_k)$, and hence the number of bands of length $n$, for any $n$ divisible by $gcd(m, m')$ is bounded below by such a function. $\square$

Now consider the example given at the end of section 4. In figure 2, we see three minimal circuits in the lower right right hand portion of the graph, labelled by $ab^{-1}a$, $ab^{-2}a$, $b^{-2}a$, and all passing through two common vertices, and three more on the left portion of the graph, labelled by $a^{-1}ba^{-1}$, $a^{-1}b^2a^{-1}$,$b^2a^{-1}$, and similarly all passing through two common vertices. We can form exponentially many longer bands by joining minimal circuits in either group of three together in various ways, as described above, finding, for example, infinitely many bands of the form $(ab^{-1}a)^\lambda(ab^{-2}a)^\mu$.

## References


[1] M C R Butler and C M Ringel, Auslander-Reiten sequences with few middle terms and applications to string algebras, Comm. Algebra 15 (1987) 145–179.

[2] W W Crawley-Boevey, Maps between representations of zero-relation algebras, Algebra 126 (1989) 259–263.

[3] J.E. Hopcroft, R. Motwani and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Second Edition, Addison Wesley, Reading, MA, 2001.

[4] J.E. Pin, *Varieties of Formal Languages*, Plenum Publishing Corp., New York, 1986.