AUTOMATA AS COALGEBRAS IN CATEGORY OF MATRICES

Nada Damljanović[1], Katarina Pavlović[1], Vučelja Lekić[1], Ljubica Đurović[2]

[1]University of Kragujevac, Faculty of Technical Sciences Čačak, Svetog Save 65, 32000 Serbia, nada.damljanovic@ftn.kg.ac.rs, katarina.pavlovic@ftn.kg.ac.rs, vucelja.lekic@ftn.kg.ac.rs
[2]University of Kragujevac, Faculty of Science, Radoja Domanovića 12, 34000 Kragujevac, Serbia, ljubica.milevic@pmf.kg.ac.rs

ORIGINAL SCIENTIFIC PAPER

Received: 17 July 2026.
Accepted: 16 August 2026.
Published: 16 November 2026.
DOI: https://doi.org/10.63395/STEDConf15022026152D289

Corresponding Author: Nada Damljanović, University of Kragujevac, Faculty of Technical Sciences Čačak, Svetog Save 65, 32000 Serbia, nada.damljanovic@ftn.kg.ac.rs

Copyright © 2026 Nada Damljanović et al.; published by UNIVERSITY PIM. This work licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.

Citation: Damljanović, N. et al. (2026). Automata as coalgebras in category of matrices. In Kojić, D. & Sredić. D. (eds.) *Conference proceedings, XV International Conference on Social and Technological Development: STED 2026* (pp. 289-296). Trebinje, B&H: University PIM.

ABSTRACT

This paper considers automata as models of systems used for information processing and computational execution. Special attention is devoted to the coalgebraic approach and the category of matrices as modern methods for the formal description of dynamic and state-based systems. By viewing automata as coalgebras, it becomes possible to model system behavior through sets of states and transitions while emphasizing behavioral equivalence. On the other hand, the category of matrices provides a natural algebraic framework for representing transitions, relations, and weights between states. The combination of these approaches establishes a unified theoretical model suitable for the analysis of finite, probabilistic, and weighted automata, as well as for applications in formal verification, communication protocols, and programming language theory. The results indicate that the coalgebraic interpretation within the category of matrices enables a deeper understanding of the structure and behavior of computational systems, thereby connecting automata theory with modern developments in computer science.

Keywords: Automata, coalgebra, category, matrix.

INTRODUCTION

Standard algebraic techniques have proven useful for describing various essential aspects of data structures used in computer science. But it is it turns out to be difficult to algebraically describe some of the dynamic structures that are widely used, especially because a notion of state which often appears can be understood in many ways. Automata or transition systems are most often used to formally describe such systems. During the last decades, there is a growing belief that it is better, instead of algebras, to represent such structures with coalgebras.

Coalgebra theory can be viewed as a framework for studying the behavior of large classes of dynamical systems, including many types of automata (deterministic, non-deterministic, probabilistic, etc.) and infinite data types (streams, trees, etc.).

Finite automata and their variants are among the fundamental structures in computer science, and their applications are wide-ranging. Deterministic finite automata are perhaps the simplest machines with finite memory (although they can represent infinite sets) and, together with finite non-deterministic automata, they represent basic examples of coalgebras.

Weighted automata [6], [10] generalize deterministic automata [1], [2] in the sense that transitions carry, in addition to an input symbol, a weight from an underlying structure and each state of these automata is assigned an output weight in that structure.

If we analyzing weighted finite automata through the lens of vector spaces and matrix operations, the behavior of a weighted finite automaton can be expressed as the product of the row vector representing the initial weights, matrices representing the weights of the transitions induced by input letters, and the column vector representing the terminal weights. Such a representation of a weighted finite automaton is called a linear representation [11].

In that sense, the set of states of a deterministic weighted automaton is equipped with a vector space structure. This formulation arises from the determinization of non-deterministic weighted automata over generic field using subset construction, a process that typically yields an infinite state set. Despite this, the algebraic properties of the linear state space enable a finite representation of these automata and efficient algorithms for their manipulation.

To formalize behavioral equivalence and construct minimization algorithms if we are using linear representation of weighted automata, it is convenient to frame them as coalgebras.

Reflecting this paradigm shift, coalgebraic theory is being adapted to quantitative contexts [3], [4], grounded by Larsen and Skou's work [5] and extended to linear weighted automata [6]. However, this framework introduces significant complexity, as it requires states to be interpreted as vectors and coalgebras as linear maps.

PRELIMINARIES

A finite weighted automaton is a quintuple $W = (A, Q, \lambda, \mu, \gamma)$, where A is the input alphabet, Q is a finite set of states, and $\lambda, \gamma: Q \to S$ are the initial and final weight functions, respectively. The function $\mu: A \to S^{(Q\times Q)}$ is defined such that $\mu(a)(p,q)$ represents the transition weight of $p \xrightarrow{a} q$. If $\mu(a)(p,q) = 0$, which means that there is no transition from state p to state q under the input symbol a. For each $a \in A, \mu(a) \in S^{(Q\times Q)}$ can be viewed as a Q-indexed matrix expressing the weights of the transitions involving the input symbol a. In a similar vein, the functions λ and γ can be regarded as Q-indexed vectors.

Underlying algebraic structure for this type of automata can be semiring $(S, +, \times, 0, 1)$ or some richer structure (for example, the field of real numbers will serve as a framework for our example).

In computer science, matrices are conventionally viewed as rectangular data structures implemented as two-dimensional arrays. On the other hand, in mathematics matrices are often treated as linear transformations. Thus, a matrix M with n rows and m columns can be defined as a function that assigns a value to each cell (r, c) of the matrix, for $1 \leq r \leq n$ and $1 \leq c \leq m$. Since the type of entries in the cells can vary, the minimal algebraic structure required for matrix operations is a semiring. Standard linear algebra typically utilizes a richer structure, primarily a field (field of real numbers serving as the canonical example).

The type of an $m \times n$ matrix M will be denoted by an arrow $m \to n$. The notation $m \xrightarrow{M} n$ (or equivalently, $n \xleftarrow{M} m$) declares the matrix M and its type. We outline some of the key operations on matrices:

– Matrix composition (multiplication) $M \cdot N$ is given by

–

$$r(M \cdot N)c = \sum_{x} \big((rMx) \times (xNc)\big),$$

where we denote the (r,c)-entry of the matrix $M \cdot N$ by $r(M \cdot N)c$, $\times$ is the multiplicative operation of the underlying semiring from which the matrix entries take their values, and $\sum$ denotes the finite iteration of its additive operation. The type of resulting matrix $M \cdot N$ is given by

$$n \xleftarrow{M} m \xleftarrow{N} k, \qquad n \xleftarrow{M \cdot N} k$$

Identity matrix of dimension n, denoted by $n \overset{id_n}{\leftarrow} n$ or $n \overset{1}{\leftarrow} n$ $(n \in \mathbb{N})$, in which all elements on the main diagonal are 1 and all off-diagonal elements are 0. Consequently, it holds that

$$id_n \cdot M = M = M \cdot id_m.$$

Transposition operation: replacing the rows of a given matrix with their corresponding columns (and vice versa). Given a matrix $n \overset{M}{\leftarrow} m$, the notation $m \overset{M^\circ}{\leftarrow} n$ refers to its transposed matrix, i.e., its converse, satisfying:

$$(M^\circ)^\circ = M,$$
$$(M \cdot N)^\circ = N^\circ \cdot M^\circ.$$

Addition: for given matrices $n \overset{M,N}{\longleftarrow} m$ of the same type, the matrix $M + N$ is defined as

$$r(M+N)c = rMc + rNc, \qquad (1 \le r \le n, 1 \le c \le m),$$

where the symbol $+$ refers to the additive operator of the underlying semiring. Furthermore, the zero matrix is defined as a matrix where each entry contains the additive identity 0 of the semiring; hence, it holds that:

$$M + 0 = M = 0 + M,$$
$$M \cdot 0 = 0 = 0 \cdot M,$$

and composition is a bilinear operation with respect to $+$, i.e.

$$M \cdot (N+P) = M \cdot N + M \cdot P,$$
$$(N+P) \cdot M = N \cdot M + P \cdot M.$$

Hadamard product: the multiplication of two matrices of the same type, denoted by $M \times N$ (or MN) is commutative, associative, and distributive with respect to addition and yields a matrix of the same type, where

$$r(MN)c = rM \cdot rNc, \quad (1 \le r \le n, 1 \le c \le m).$$

The identity element for this product is the matrix whose entries all equal 1 (the multiplicative identity of the underlying semiring).

Together with the composition operation and the identity element, matrices form a category whose objects are the matrix dimensions and whose morphisms $(m \overset{M}{\leftarrow} n)$ are the matrices themselves [8], [9]. We will denote this category by Mat_S, parameterized by a semiring S or some other (richer) algebraic structure. Also, vectors can be viewed as a special case of matrices where one of the dimensions is 1.

AUTOMATA AND CATEGORY OF MATRICES

Due to the commutative property of the additive operation in the semiring S, the objects in the matrix category Mat_S, namely the matrix types, can be generalized from numerical values $n, m \in \mathbb{N}_0$ to arbitrary non-numerical sets (types) (A, B). This is achieved by replacing $m + n$

with the disjoint union $A + B$, mn with the Cartesian product $A \times B$, 1 (the multiplicative identity of the semiring S) with a singleton set, 0 (the additive identity of the semiring S) with the empty set $\emptyset$, and so forth. Conversely, a dimension n corresponds to the type $\{0, 1, 2, \dots, n-1\}$.

So, we can define the weighted automaton in [6] as a quadruple $W = (Q, A, \mu, \gamma)$, where the initial weight function λ is omitted. In this case, the final weight function γ is assigned the type $Q \to 1$, which is now considered a row vector in the category Mat_S. The transition function μ can be seen in two ways: either as a matrix of type $Q \times A \to Q$ or as a matrix of type $Q \to Q \times A$. In the category Mat_S, these two types are isomorphic, and since the representation $Q \to Q \times A$ of the function μ is coalgebraic, the weighted automaton in the category Mat_S can be defined by the following diagram

$$Q \times A \overset{\mu}{\leftarrow} Q \overset{\gamma}{\to} 1.$$

Since every matrix can be partitioned into corresponding matrix blocks (submatrices), there are two ways to construct a new matrix from two given matrices, say M and N:

1. The matrix $[M \mid N]$ is formed by appending the columns of N to the columns of M;
2. The matrix $\left[\frac{M}{N}\right]$ is formed by stacking the matrix M on top of the matrix N.

A detailed description of both constructions can be found in [7].

Now, the functions μ and γ of the weighted automaton $W = (Q, A, \mu, \gamma)$ can be viewed as a single matrix of type

$$(Q \times A) + 1 \overset{W}{\leftarrow} Q,$$

where

$$W = \left[\frac{\mu}{\gamma}\right].$$

In other words, a weighted automaton represents a coalgebra of the functor

$$FX = (X \otimes id) \oplus id,$$

in the category Mat_S, where $\otimes$ denotes the Kronecker product of matrices.

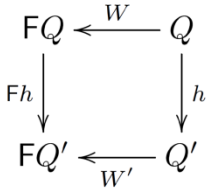
To define the basic coalgebraic concepts in the category Mat_S, every function $A \overset{f}{\to} B$ must be represented by a matrix $[[f]]$ of the same type, defined by

$$b[[f]]a \ (b =_S f(a)),$$

where $b =_S x$ equals the multiplicative identity 1 of the semiring S if $y = x$, and 0 otherwise.

Based on this, the matrix $[[f]]$ is a representation of the graph of the function f: an entry of the matrix $[[f]]$ is equal to 1 for $(f(a), a)$ and 0 otherwise.

For the functor $FX = (X \otimes id) \oplus id$, an F-coalgebra homomorphism (or F-cohomomorphism) between two weighted automata W and W' is a function h for which the following diagram commutes:



In other words, the following holds

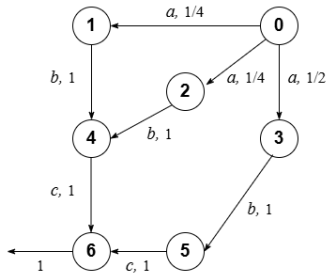
$$(Fh)\cdot W = W'\cdot h \Leftrightarrow \begin{cases}(h\otimes id)\cdot \mu = \mu'\cdot h,\\ \gamma = \gamma'\cdot h.\end{cases}$$

The identity $(h\otimes id)\cdot\mu = \mu'\cdot h$, for matrices of type $Q'\times A \leftarrow Q$, means that the weight corresponding to the transition $q' \overset{a}{\leftarrow} h(q)$ in the target weighted automaton is the accumulation of the weights of all transitions $p \overset{a}{\leftarrow} q$ in the source automaton, for every p such that $h(p) = q'$.

The second identity, $\gamma = \gamma'\cdot h$, implies that $\gamma(q) = \gamma'(h(q))$ when γ and γ' are viewed as functions again.

EXAMPLE

Let us consider the weighted automaton $W = (Q, A, \mu, \gamma)$ from the figure, where $Q = \{0, 1, 2, 3, 4, 5, 6\}$ and $A = \{a, b\}$.



The matrix μ is given by the following table, where the columns are indexed by the elements of Q and the rows by the Cartesian product $Q\times A$. Transitions that do not lead to any state and are labeled with 1 correspond to the row vector $\gamma = (0\,0\,0\,0\,0\,0\,1)$ of type $1\leftarrow Q$.

Q	A	0	1	2	3	4	5	6	Q	A	0	1	2	3	4	5	6
0	a	0	0	0	0	0	0	0	4	a	0	0	0	0	0	0	0
0	b	0	0	0	0	0	0	0	4	b	0	1	1	0	0	0	0
0	c	0	0	0	0	0	0	0	4	c	0	0	0	0	0	0	0
1	a	$\frac{1}{4}$	0	0	0	0	0	0	5	a	0	0	0	0	0	0	0
1	b	0	0	0	0	0	0	0	5	b	0	0	0	1	0	0	0
1	c	0	0	0	0	0	0	0	5	c	0	0	0	0	0	0	0
2	a	$\frac{1}{4}$	0	0	0	0	0	0	6	a	0	0	0	0	0	0	0
2	b	0	0	0	0	0	0	0	6	b	0	0	0	0	0	0	0
2	c	0	0	0	0	0	0	0	6	c	0	0	0	0	1	1	0
3	a	$\frac{1}{2}$	0	0	0	0	0	0									
3	b	0	0	0	0	0	0	0									
3	c	0	0	0	0	0	0	0									

Figure 1. Transition matrix μ of automaton

Is the equivalence relation K shown in the following figure a bisimulation relation?

K	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0
2	0	1	1	0	0	0	0
3	0	0	0	1	0	0	0
4	0	0	0	0	1	1	0
5	0	0	0	0	1	1	0
6	0	0	0	0	0	0	1

Figure 2. Equivalence relation K

The relation K has five equivalence classes, which can be represented by a quotient automaton with a new state set $Q' = \{0, I, II, III, IV\}$, using the corresponding surjective homomorphism $Q' \overset{h}{\leftarrow} Q$.

h	0	I	II	III	IV
0	1	0	0	0	0
1	0	1	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	1	0
6	0	0	0	0	1

Figure 3. Homomorphism $h: Q' \leftarrow Q$

From [6], the equivalence relation $Q \overset{K}{\leftarrow} Q$ is a bisimulation for a given weighted automaton, if $K = h^{\circ} \cdot h$ and

$$FK \cdot W = FK \cdot W \cdot K^{*},$$

where $K^{*} = h^{\bullet} \cdot h$, and $h^{\bullet}$ is inverse of homomorphism h.

It is easy to check that for given relation K and $h^{\bullet} = h^{\circ} \cdot (h \cdot h^{\circ})^{-1}$, obtained by inversion of diagonal $h \cdot h^{\circ}$, holds

$$(K \otimes id) \cdot \mu = (K \otimes id) \cdot \mu \cdot K^{*},$$
$$\gamma = \gamma \cdot K^{*}.$$

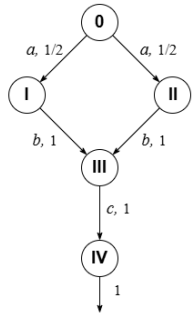
Quotient automaton $W' = W_{/K} = (Fh) \cdot W \cdot h^{\bullet}$, is defined by

$$\mu' = \mu_{/K} = (h \otimes id) \cdot \mu \cdot h^{\bullet},$$
$$\gamma' = \gamma_{/K} = \gamma' \cdot h^{\bullet}.$$

Q'	A	0	I	II	III	IV
0	a	0	0	0	0	0
0	b	0	0	0	0	0
0	c	0	0	0	0	0
I	a	$\frac{1}{2}$	0	0	0	0
I	b	0	0	0	0	0
I	c	0	0	0	0	0
II	a	$\frac{1}{2}$	0	0	0	0
II	b	0	0	0	0	0
II	c	0	0	0	0	0
III	a	0	0	0	0	0
III	b	0	1	1	0	0
III	c	0	0	0	0	0
IV	a	0	0	0	0	0
IV	b	0	0	0	0	0
IV	c	0	0	0	1	0

Figure 4. Transition table μ' of quotient automaton

We also give transition graph of a quotient automaton Q' :



CONCLUSION

Category of matrices can be viewed as a natural evolution of the category of sets, because enables a coalgebraic approach to weighted automata. The paper highlights the importance of the representation of weighted automata in a coalgebraic setting, and through our example, we illustrate the procedure of computing an equivalence relation which represents a bisimulation for the given weighted automaton. The obtained quotient automaton simulates the original automaton and obviously features a smaller state space.

ACKNOWLEDGEMENTS

This paper is supported by the Ministry of Science and Technological Development of Serbia, Grant No. 451-03-34/2026-03/200132.

We would like to express our special gratitude to the editors and reviewers for their time and insightful comments.

DECLARATIONS OF INTEREST STATEMENT

The authors affirm that there are no conflicts of interest to declare in relation to the research presented in this paper.

LITERATURE

1. Schützenberger, M. P. (1961). On the definition of a family of automata. *Information and control*, *4*(2-3), 245-270.
2. Droste, M., & Gastin, P. (2005). Weighted automata and weighted logics. In *International Colloquium on Automata, Languages, and Programming* (pp. 513-525). Berlin, Heidelberg: Springer Berlin Heidelberg.
3. Sokolova, A. (2005). *Coalgebraic Analysis of Probabilistic Systems* [Ph.D. dissertation, Eindhoven University of Technology]. Eindhoven, The Netherlands.
4. Silva, A., Bonchi, A. F., Bonsangue, M. & Rutten, J. (2011). Quantitative Kleene coalgebras. *Information and Computation, 209*(5), 822-849.
5. Larsen, K. & Skou, A. (1991). Bisimulation through probabilistic testing. *Information and Computation, 94*(1), 1-28.
6. Bonchi, F., Bonsangue, M., Boreale, M., Rutten, J. & Silva, A. (2012). A coalgebraic perspective on linear weighted automata. *Information and Computation, 211*, 77-105.
7. Oliveira, J.N. (2013). Weighted automata as coalgebras in categories of matrices. *International Journal of Foundations of Computer Science, 24*(6), 709-728.
8. MacLane, S. (1971). *Categories for the Working Mathematician.* Springer.
9. MacLane, S. & Birkhoff, G. (1999). *Algebra*. AMS Chelsea.
10. Buchholz, P. (2008). Bisimulation relations for weighted automata. *Theoretical Computer Science, 393*(1-3), 109-123.
11. Damljanović N., Ćirić M. & Ignjatović J. (2023). Weighted Automata over Vector Spaces. *16th International Conference on Automata and Formal Languages (AFL 2023) EPTCS 386*, 67-81.