Proceedings of the Conference on March 14 - International Day of Mathematics

Vuković, Mirjana, urednik; Nurkanović, Mehmed, urednik

2024-12-26

Academy of Sciences and Arts of Bosnia and Herzegovina

https://bastina.anubih.ba/handle/123456789/798

PROCEEDINGS OF THE CONFERENCE ON MARCH 14 – INTERNATIONAL DAY OF MATHEMATICS
Special Editions ANUBiH, Book CCXVI, OPMN Book 30
Sarajevo 2024, **p. 90–99** DOI: 10.5644/PI2024.216.11

USING DIRECTED GRAPHS TO DESCRIBE AUTOMATION PROCESSES AND ANALYZE CONTROL SYSTEMS

IVANA ZUBAC, SNJEŽANA REZIĆ, AND JADRANKO BATISTA

Dedicated to the 75th birthday of our dear Professor Mirjana Vuković

ABSTRACT. In many ways, engineering has always been dependent on mathematical background. This was usually mathematical analysis. However, many problems in natural, technical and social science can be successfully formulated in terms of graph theory. Today graph theory is well developed, strongly stimulated by technical and chemical applications. Graphs have been increasingly used in many fields, such as automation, programming or algorithms. In the automation process directed graphs are used for description and analysis of the control system, where nodes represent states and arrows direct edges or transitions between states automation that manages processes. In this paper we show how to use graphs to simplify the process of automation.

1. INTRODUCTION

Many problems in the natural, technical and social sciences can be successfully formulated in terms of graph theory. Today, graph theory is well developed, strongly motivated by technical and chemical applications, and it has established itself as an important mathematical tool in a wide variety of subjects, from operational research to genetics and linguistics, and from electrical and mechanical engineering and geography to sociology and architecture. For general background on graph theory and terminology, we refer the reader to the classic book by West D.B. [7], [4]. For the theory of directed graphs or digraphs, which is not defined here, we also recommend [1], [3]. Graph theory studies the ways in which sets of points, called vertices, can be connected by lines or arcs, called edges. The term graph in this context differs from graphs that show mathematical relationships and functions in coordinate systems. A directed graph has oriented edges, which we show with arrows. In practice, there is a great need to display various systems using such graphs (e.g. traffic regulation in a city, liquid flow in a system, transport of goods, process automation, etc.)

1.1. Basic definition of graphs and digraphs

A graph G consists of a finite nonempty set V of p vertices together with a defined set X of q unordered pairs of distinct vertices of V. Each pair $x = \{u, v\}$ of points in X is an edge of G, and we say that x joins u and v. A graph with p vertices and q edges is called a (p, q) graph.

2020 *Mathematics Subject Classification.* 05C20, 05C90, 94C15.
Key words and phrases. graph theory; digraph; automation; matrix.

A directed graph, or **digraph** D consists of a non-empty finite set $V(D)$ of elements called vertices, and a finite family $A(D)$ of ordered pairs of elements of $V(D)$ called arcs arc or directed edge. We call $V(D)$ the vertex set and $A(D)$ the arc family of D. An arc (v,w) is usually abbreviated to vw, the ordering of the vertices in an arc being indicated by an arrow. If D is a digraph, the graph obtained from D by 'removing the arrows' (that is, by replacing each arc of the form vw by a corresponding edge vw) is the underlying graph of D.

Two or more arcs with the same beginning and end are called multiple arcs. So, D is the orientation of G and we write $D = \overrightarrow{G}$. A directed graph consists of a finite set of V vertices (vertices) and a collection of ordered pairs. A strict graph is a directed graph whose associated graph is simple and complete. The associated graph $D(G)$ of a graph G is a directed graph obtained from G by replacing each edge with two oppositely oriented arcs with the same endpoints. The associated graph $G(D)$ of a directed graph D is the graph obtained from D by deleting all arrows. A tournament is a directed graph whose graph is simple and complete.

Many terms used in graphs are also used in directed graphs. For example directed cycle or dicycle, directed path or dipath. A directed graph is acyclic if it does not contain a dicycle.

On the other hand, oriented edges between two vertices are allowed in simple directed graphs, and such a pair of edges is called a digon. We say that a digraph D is weakly connected (or connected for short) if the associated graph is connected, and strongly connected if for every two vertices $u, v \in V(D)$ there is a (u,v)-diput.

A component of a digraph is a connectivity component of an associated graph.

Unlike graphs, digraphs have two types of vertex degrees. The *in-degree* of a vertex in a digraph is the number of arrows coming into it, and similarly its *out-degree* is the number of arrows out of it. More precisely, for $v \in V(D)$ we define:

- **indeg(v),** $d_D^-(v)$ and
- **outdeg(v)** $d_D^+(v)$.

The following proposition is analogous to the proposition about the degree of an undirected graph:

Proposition 1.1. *Let D be a directed graph with a set of vertices $V(D)$ and a set of arcs $A(D)$. Then*

$$\sum_{v \in V(D)} d^-(v) = |A(D)| = \sum_{v \in V(D)} d^+(v).$$

In this paper, all directed graphs are finite and can have loops and multiple edges (edges with the same starting and ending vertices). A directed graph D is simple if D has no loops and there is at most one edge from v_i to v_j for any $v_i, v_j \in V(D)$.

1.2. **Matrix representation of a graph**

It is common to represent a graph using a graphical and matrix display. The graph is completely determined by adjacency or incidence matrices. The adjacency matrix $A = \left[a_{ij}\right]$ of a labeled graph G with p points is a $p \times p$ matrix in which $a_{ij} = 1$ if v_i is adjacent to v_j and $a_{ij} = 0$ otherwise. The second matrix, associated with the graph G in which vertices and edges are marked, is the incidence matrix $B = \left[b_{ij}\right]$. This $p \times q$ matrix has $b_{ij} = 1$ if v_i and x_j are incident and $b_{ij} = 0$ otherwise.

The adjacency matrix of the directed graph D on the set of vertices $V(D) = \{v_1, v_2, \ldots, v_n\}$ is the square matrix $A(D) = \left[a_{ij}\right]$ of order n, where a_{ij} is the number of arcs in D starting at v_i and ending at v_j .

If there are no multiple arcs, then the elements of the adjacency matrix are only zeros and ones, otherwise they are non-negative integers. Each adjacency matrix uniquely defines a directed graph D.

It is often possible to use these matrices to identify certain properties of a graph. For a digraph, the degree of a vertex is the sum of the column and row entries corresponding to its adjacency matrix. Row values corresponding to the vertex v represent edges with v as the starting vertex, and column values corresponding to v represent edges with v as the final vertex. The sum of all entries in the adjacency matrix is, of course, the total number of edges in the digraph.

A binary relation on a set can be represented by a digraph. Let R be a binary relation on the set A, that is, R is a subset of $A \times A$. Then the digraph representing R can be constructed as follows:

Let the elements of A be vertices of a digraph G, and let $\langle x, y\rangle$ be an arc of G from vertex x to vertex y if and only if $\langle x, y\rangle$ is in R.

2. AUTOMATION PROCESS

Automation of the process is replacing human labor with machines, not only in terms of strength, but also intellectual work. Technically, the automatic machine (technical system) consists of three groups of elements:

- senses (sensors, cameras, microphones, etc.)
- controllers (processors who process information)
- executive elements.

For theoretical background on of automation of the process not defined here we also recommend [2], [5], [6]. *The management process* is a system, in which one or more input variables, over the legality which characterizes this system, affect other variables as output values. *The process* is quantitative and/or qualitative change dependent on the weather, and it takes place in nature, society and technology.

Automating processes involve integrating disparate systems for seamless data flow across an organization. Additionally, it's a critical part of continuous optimization, and core to many organization's digital transformation initiatives.

Information is data about a particular phenomenon, concept or event. The holder of information is the signal. The signal is a changeable and measurable variable at the entry or exit of the system, and can take a variety of physical forms. Classification of signals depends primarily on the respective parameters of amplitude and time. We distinguish: continuous, discrete and binary signals. Digital automat is a universal sequential circuit, whose behaviour depends only on the current and previous input data-events. Working machines can be explained by the theory of systems management.

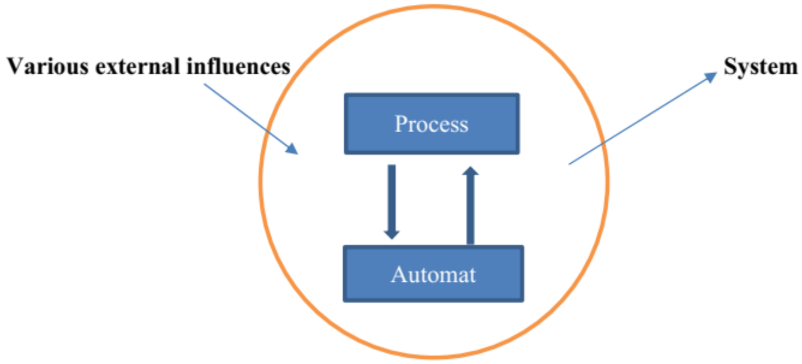

Figure 1. Automation process

An automat measures the state of the process and resolves all its essential conditions. On the basis of the current and previous events in the process we can determine the optimal action or series of actions. These actions should result in an automat process in the optimal mode. To make this possible, the automat must have a sufficient set of actions, in order to compensate for any predictable impact of the environment. We say that the process must be controllable, so that the management could work. If the value of the output variable depends not only on the current values of the input variable, but also on the past values (total input-output) we says that the digital system is a sequential system or is automatic.

The properties of automata:

- *Define finality (there are a finite number of states, the final memory);*
- *Define discretion (working in discrete time);*
- *Define the digital mode (available digital inputs and outputs);*
- *Define determination (unambiguously perform its function);*
- *Define the specificity (completely - all possible sequences of input events - expects an arbitrary set of inputs; incomplete - possible are only a series of input events);*
- *Define synchronicity (discrete time defined by phased signal).*

3. AUTOMATION OF THE SELF-SERVICE DEVICE FOR BEVERAGES

Before starting the design of the automation process, it is important to determine the inputs and outputs of the automaton and state automata.

We will assume that the self-service machine dispenses four drinks, which means that we need to have four inputs (select1, select2, select3, select4) for the drinks. We also assume that the machine only accepts 0.50 KM coins and one convertible mark, which we mark as km_50 and km_100. We must also implement an input for canceling the order if the customer wants to return the money. Return, drink and change will represent our exits. A refund returns the money back to the person, a drink out throws out the selected drink, while a change returns the excess money.

In the juice vending machine, one bottle costs 1.50 KM. The operation of the machine is controlled by a sequential circuit with two inputs X_1 and X_2.

A logic unit appears at input X_1 (for the duration of one clock pulse) when a 1 KM coin is inserted, and a logic unit appears at input X_2 (for the same duration) when a 2 KM coin is inserted. At all other times, inputs X_1 and X_2 are at logic zero. The sequential circuit has two outputs Y_1 and Y_2. The logic unit at output Y_1 starts the motor that ejects the bottle of juice, while the logic unit at output Y_2 orders the device to eject change of 0.50 KM.

It's apparently a sequential circuit, since the device has to keep track of how much money was previously inserted in order to know how to react to the insertion of the next coin. It is important to note that it is not necessary to keep records of the total amount of money in the device, but only of the amount of money that was inserted after the last ejected bottle of juice. Namely, after each ejected bottle, the device, from the user's point of view, behaves as if it had just started working. This fact is of crucial importance, because based on it we can conclude that, from the aspect of behavior, the assembly can only be in one of three different states:
S_0 - No coins were inserted after the last bottle was thrown
S_1 - After throwing out the last bottle, a total of 0.50 KM was put
S_2 - After throwing out the last bottle, a total of 1 KM was put

The bottle is ejected if a 1 KM coin is inserted in state S_1, or if either a 0.50 KM coin or a 1 KM coin is inserted in state S_2 (in the latter case, change is also thrown out). In doing so, the device returns to the initial state S_0 .

Based on previous thinking and analysis, it is quite easy to draw a graph that describes the circuit's operation.

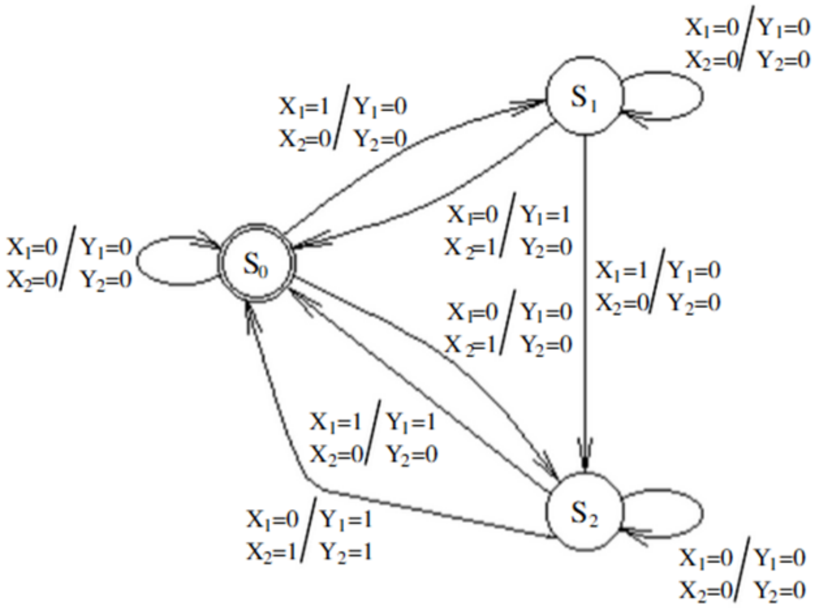

Figure 2. Automaton graph

Alternatively, instead of a graph, the operation of the circuit can be described with the following transition table, which is easy to compile by directly reading the graph.

TABLE 1. Transition table

X_1	X_2	The old state	The new state	Y_1	Y_2
0	0	S_0	S_0	0	0
0	0	S_1	S_1	0	0
0	0	S_2	S_2	0	0
0	1	S_0	S_2	0	0
0	1	S_1	S_0	1	0
0	1	S_2	S_0	1	1
1	0	S_0	S_1	0	0
1	0	S_1	S_2	0	0
1	0	S_2	S_0	1	0

The displayed states must be coded with binary numbers. Let's adopt the following encoding:

TABLE 2. Table of code

X_1	X_2	$Q_1(n)$	$Q_2(n)$	$Q_1(n+1)$	$Q_2(n+1)$	Y_1	Y_2
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	x	x	x	x
0	0	1	1	1	1	0	0
0	1	0	0	1	1	0	0
0	1	0	1	0	0	1	0
0	1	1	0	x	x	x	x
0	1	1	1	0	0	1	1
1	0	0	0	0	1	0	0
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	0	0	1	0
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

To implement the transition/output table, JK and D bistable will be used for memory and state change.

We can see the implementation of the automaton using combination circuits in Figure 3.

3.1. **Logical scheme**

In order to simplify the entire cycle of buying a beverage, let's look at the flow diagram of our device in Figure 4.

The state diagram consists of four states (User's selection, Waiting for the money to be entered, ejecting the product and servicing if the selection is not available). Primarily,

TABLE 3. Table of transition/exits

X_1	X_2	$Q_1(n)$	$Q_2(n)$	$Q_1(n+1)$	$Q_2(n+1)$	Y_1	Y_2	J_1	K_1	D_2
0	0	0	0	0	0	0	0	0	x	0
0	0	0	1	0	1	0	0	0	x	1
0	0	1	0	x	x	x	x	x	x	x
0	0	1	1	1	1	0	0	x	0	1
0	1	0	0	1	1	0	0	1	x	1
0	1	0	1	0	0	1	0	0	x	0
0	1	1	0	x	x	x	x	x	x	x
0	1	1	1	0	0	1	1	x	1	0
1	0	0	0	0	1	0	0	0	x	1
1	0	0	1	1	1	0	0	1	x	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	0	0	1	0	x	1	0
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

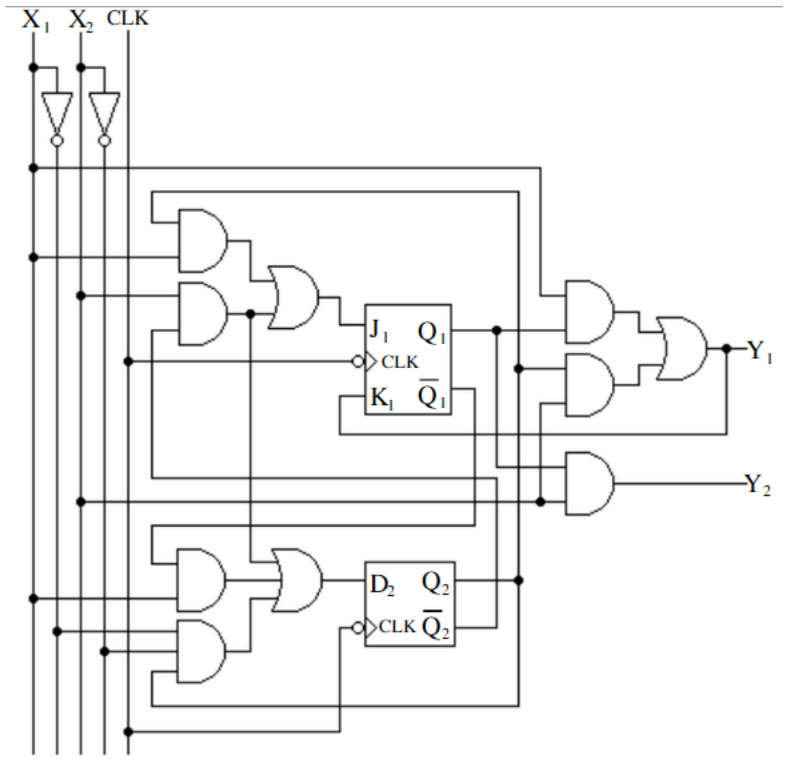

Figure 3. Logical scheme

when the reset button is pressed, the device will be ready to select a beverage. After this state the user chooses a drink, this state can be one of the four listed selections. The device accepts two types of coins: 0.50 KM and 1 KM. Let's assume that the useful one

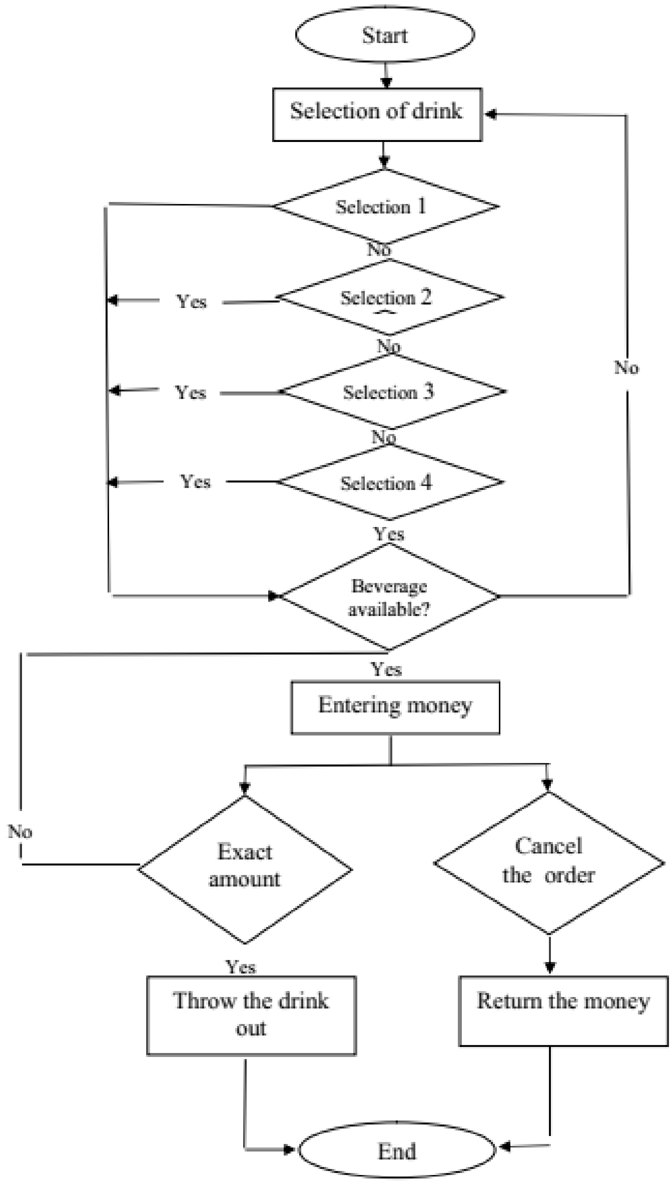

Figure 4. Flow diagram

chooses the drink selection1. The device primarily checks whether the drink is available or not. After that, the control unit switches to another state, where it waits for the money to be entered. If a 0.50 KM coin is entered, it goes to state_1 where it waits until the desired amount is entered. If one convertible mark is entered, it switches to state_2 and waits until one and a half convertible marks are entered. When the desired amount of money is filled, the device ejects the drink.

The obtained graph of the operation of the automaton clearly shows the connection between all its internal states and transitions, i.e. the flow of information and signals. All this enables us to describe and analyze the entire management system, but also to facilitate the implementation and simulation of theoretical and real systems, or their synthesis.

Among the classical methods of mathematical description of the control system and graphoanalytical methods, there is a correlation of both models, and the developed graphoanalytical methods easily lead to a complex mathematical model, which significantly shortens the time of setting up the model, synthesizing the control and developing the control algorithm. Also, we usually write graphs in the computer memory via the adjacency matrix, which enables easier management and monitoring of the automated process.

4. Conclusion

Unlike conventional methods of mathematical modeling and mathematical description of automatic control systems, graphoanalytical methods have taken precedence not only in practice but also in research, because they lead to a simpler and more comprehensible description of the system, and thus to its analysis and ultimately to the synthesis of optimal solutions.

The fact that an edge connects node A to node B without requiring feedback is the reason for using a directed graph in the development of automation for control processes, because feedback is not always needed in the description, analysis and synthesis of automation for process control.

References

[1] J. Bang-Jensen, G. Gutin, *Digraphs Theory, Algorithms and Applications*, Springer-Verlag, London, 2007.

[2] M. Balach, *Complete Digital Design*, McGraw-Hill, New York, 2003.

[3] R. Garnier, J. Taylor, *Discrete Mathematics for New Technology*, Institute of Physics Publishing, Bristol and Philadelphia, 2002.

[4] F. Harary, *Graph theory*, Adison Wesley Longman Publishing Co., Massachusetts, 1972.

[5] J. Hopcroft, J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Boston, 1979.

[6] Edited by A. D. Rodic, *Automation Control – Theory and Practice*, InTech, 2009.

[7] D. B. West, *Introduction to Graph Theory*, Prentice Hall Inc., Upper Saddle River, NJ, 2007.

(Received: May 17, 2024)
(Revised: September 01, 2024)

Ivana Zubac
University of Mostar
Faculty of Mechanical Engineering, Computing and Electrical Engineering,
Matice hrvatske b.b.
88000 Mostar, BiH
e-mail: *ivana.zubac@fsre.sum.ba*
and
Snježana Rezić
University of Mostar
Faculty of Mechanical Engineering, Computing and Electrical Engineering,
Matice hrvatske b.b.
88000 Mostar, BiH
e-mail: *snjezana.rezic@fsre.sum.ba*
and
Jadranko Batista
University of Mostar
Faculty of Science and Education
Matice hrvatske b.b.
88000 Mostar, BiH
e-mail: *jadranko.batista@fpmoz.sum.ba*