Communications in Mathematics and Applications
Vol. 13, No. 1, pp. 183–197, 2022
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v13i1.1704

http://www.rgnpublications.com

Research Article

Automata in Chinese Remainder Theorem

Mridul Dutta*[1] and Helen K. Saikia[2]

[1] *Department of Mathematics, Dudhnoi College, Goalpara, Assam, India*
[2] *Department of Mathematics, Gauhati University, Assam, India*
***Corresponding author:** mridulduttamc@gmail.com

Received: October 20, 2021 **Accepted:** December 13, 2021

Abstract. Automaton is a system that spontaneously gives an output from an input. The input may be energy, information, materials, etc. The system works without the intervention of man. Simply automaton (plural: automata or automatons) is a self–operating machine. Its synonym is ROBOT. In this paper, an attempt has been made to exhibit the relation between linear congruence and automata theory. Also, an effort has been put to solve certain problems of the Chinese Remainder theorem using the Cartesian product of finite automata theory. In deterministic finite automata, the acceptable strings give the solutions of the Chinese Remainder Theorem (CRT). The main result of the paper is that residue classes can be recognized by finite automaton. This is the novelty of the article. Finally, we conclude with certain examples and non-examples alike!

Keywords. Automata, Cartesian product of DFA, Chinese Remainder Theorem

Mathematics Subject Classification (2020). 11T71, 11Y40, 68Q70

1. Introduction

Automaton (in plural Automata) is an abstract self-operating machine which follows a predetermined sequence of operation that automatically gives an output from an input. Here input may be energy, information, materials, etc. The system works without the intervention of man. Automata theory plays a huge essential role in applied areas. The really important areas encompass communication, transportation, health care, electronic banking, etc. Mainly, finite automata are significant in several different areas, including electrical engineering, linguistics, computer science, philosophy, biology, mathematics, etc. In computer science, automata have

been extensively utilized in text processing, compilers, software and hardware design, network protocol, etc. ([8,20]).

Number theory is a topic of pure mathematics mainly concern with the learning of natural numbers and the integers. Number theory is partly experimental and partly theoretical. Number theory belongs to the purest of pure mathematics. The number theory, as such, is less utilized in automata theory in contrast to calculus, geometry, Numerical analysis, algebra, computing, and so on. The issue was that it could not be applied directly in any application. But, the number theory, merged with the computational power of Computer science, gives interesting results to day-to-day life problems. The top familiar application of Number theory in public-key cryptography, such as the RSA algorithm ([1]).

Many researchers have carried out their research works on Number theory and Automata theory. Researchers like Adamczewski and Bell [1], Glushkov [6], Muller [13], Wolfram [22], and Perrin *et al*. [14] have developed the theory considerably. Authors such as Hartmanis and Shank [7], Rauzy [17], Restivo [18], Pin [15], Salomaa [20], Ding *et al*. [4], Allouch [2], and Hsieh [10] have done considerable work (between 1965-2000) on various aspects of the recognition of primes by automata, application of finite automata to Number theory and to the theory of codes, Chinese remainder theorem with application in computing, coding, cryptography, cellular automata, etc. Moreover recently, the mathematician Rigo [19], Rajasekaran *et al*. [16], and Steiner [21], etc. have relevant their extensive works with formal language, automata, and numerical systems, additive number theory via automata theory, etc.

2. Preliminaries

Mathematics is progressively accepted as a significant apparatus to study multiplex systems. It helps us to discuss, understand, and develop several fields in real life. Mainly all branches of mathematics are very convenient in computer science comprising algebra, calculus, number theory, topology, biology, physics, etc. In this article, an attempt is made to study some results by utilizing the tools of number theory and automaton theory.

2.1 Mathematical definition of Deterministic Finite Automata (DFA)

A *Deterministic Finite Automata* (DFA)[6, 9, 12] can be formally defined as a 5-tuple $\sum = (Q, A, \delta, q_0^*, F)$ where $Q(\neq \phi)$ is a finite set of states, A is a finite non-empty set of inputs, $\delta : Q \times A \to Q$ is defined by $\delta(q_0^*, a) = q_1$; $q_0^*, q_1 \in Q$, $a \in A$, q_0^* is the initial state, F is the set of final states and $F \subseteq Q$. A string x is accepted by finite state automata $\sum = (Q, A, \delta, q_0^*, F)$ if $\delta(q_0^*, x) = p$ for some $p \in F$. A final state is also called an accepting state. The initial state is denoted by an arrow mark and the final state is denoted by a double circle. The input is accepted when all input is read and match by transitions and the automaton is in a final state. Also, the table which represents that list of transition function (rules) of a finite automaton is called the transition table. A finite-state automaton is a machine that constructs computing by reading a one-way read-only tape. The input is produced up of 'words' written on the tape. The written words use a describe alphabet which is called the input alphabet and the words create a string. The Finite automata will be produced up of the input-output relations at every

state and also the modifications of the states that will appear in receiving the input at a particular state. At the end of the process, it becomes visible whether the input is accepted or rejected by the automaton machine. Also, Deterministic refers to the distinctiveness of the computation. The finite automata are called deterministic finite automata if the machine reads an input string one symbol at a time.

2.2 Cartesian product of Finite Automata

If $\sum_1 = (Q_1, A_1, q_0', \delta_1, F_1)$ and $\sum_2 = (Q_2, A_2, q_0'', \delta_2, F_2)$ are two finite automata then the Cartesian product [5, 10] of $\sum_1$ and $\sum_2$ is $\sum = \sum_1 \times \sum_2$ is given by $\sum = (Q, A, \delta, q_0^*, F)$ where $Q = Q_1 \times Q_2$, $A = A_1 = A_2$, $q_0^* = \{(q_0', q_0'')\}$, $(p_i, q_j) \in F$ iff $p_i \in F_1$, $q_j \in F_2$, $\delta = \delta_1 \times \delta_2$ with transition function $\delta : (Q_1 \times Q_2) \times A \to Q_1 \times Q_2$ is defined by $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$, $q_1 \in Q_1$, $q_2 \in Q_2$.

2.3 Statement of Chinese Remainder Theorem

If $m_1, m_2, m_3, \ldots, m_k$ are pairwise relatively prime positive integers, and if $a_1, a_2, a_3, \ldots, a_k$ are any integers, then the simultaneous congruences $x \equiv a_1 (\text{mod } m_1)$, $x \equiv a_2 (\text{mod } m_2), \ldots, x \equiv a_k (\text{mod } m_k)$ have a solution, and the solution is unique modulo M, where $M = m_1 \cdot m_2 \cdot m_3 \cdot \ldots \cdot m_k$ [3].

In the system of numeration each number is depicted by its base. If the base is 2 it is a binary number, if the base is 8 it is an octal number, if the base is 10, then it is called the decimal number, and so on. The conversion of decimal numbers to any other number system is an easy method. The principal motivation of this work is to utilizing automata structure, i.e., transition table, state figure to solve some important theorems which naturally arise in number theory. A study of the problems of CRT by finite automata is presented in this paper.

3. Main Results

A well-designed automata method can go easy of mundane and unnecessary manual tasks with speedy, efficiency reduce test cost of maintenance with lower risks. In this section, an automata theory is used with a new idea on *Chinese Remainder Theorems* (CRT). An effort is made to fit automata which are really obliging to traverse the characteristic on CRT of Number theory.

Theorem 3.1. *The binary representation of an unsigned decimal integer which is divided by n, $n \in \mathbb{N}$ gives a set of all acceptable binary strings of deterministic finite automata, i.e., the binary representation of the congruence $x \equiv 0 (\text{mod } n)$, $x, n \in \mathbb{N}$ gives us a set of all acceptable binary strings of DFA* [11].

Proof. When a number is divided by n, the possible remainders are $0, 1, 2, \ldots, n-1$. So, we can assign each state as $q_0, q_1, q_2, \ldots, q_{n-1}$. Also, we know that, for any binary string, if we add a bit at *Least Significant Bit* (LSB), then the previous value becomes twice. (Let us consider a string 1001. The decimal value is 9, if we add another 1 at LSB, the string becomes 10010 then the decimal value becomes 18.) Generally, we can write that n becomes $2n + b$, where n the previous number and b is the added bit. So $(2n + b)(\text{mod } n) = 2n(\text{mod } n) + b(\text{mod } n)$. As b is either 0 or 1, $b(\text{mod } n) = b$, $2n(\text{mod } n)$ is any one of $0, 1, 2, \ldots, n-1$, i.e., 2(state number)+a.

For this, we consider $\sum = (Q, A, \delta, q_0^*, F)$, where $Q = \{q_0, q_1, q_2, \ldots, q_{n-1}\}$, $A = \{0,1\}$, $q_0^* = F = \{q_0\}$ and state transition Table 1 and the state transition Diagram 1.

Table 1. State transition table of $\sum$

δ Input → / States ↓	0	1
q_0	$2.0+0=0$ means q_0	$2.0+1=1$ means q_1
q_1	$2.1+0=2$ means q_2	$2.1+1=3$ means q_3
q_2	$2.2+0=4$ means q_4	$2.2+1=5$ means q_5
q_3	$2.3+0=6$ means q_6	$2.3+1=7$ means q_7
…	…	…
q_{n-2}	$2.(n-2)+0=(2n-4)(\text{mod } n)=n-4$ means q_{n-4}	$2.(n-2)+1=(2n-3)(\text{mod } n)=n-3$ means q_{n-3}
q_{n-1}	$2.(n-1)+0=(2n-2)(\text{mod } n)=n-2$ means q_{n-2}	$2.(n-1)+1=(2n-1)(\text{mod } n)=n-1$ means q_{n-1}

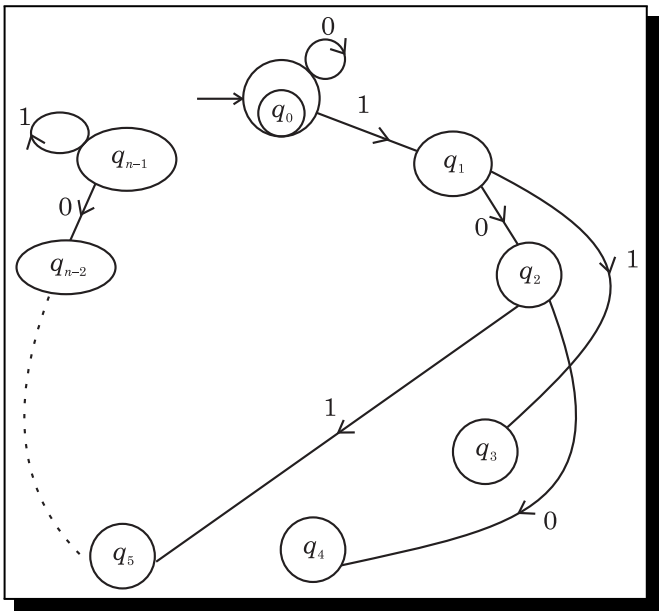

Diagram 1. A DFA for divisible by n

Therefore, we see in the diagram that the collection of those acceptable binary strings which gives the final state of the automata represent the solution of the linear congruence in decimal systems. Hence, a linear congruence $x \equiv 0(\text{mod } n)$, $x, n \in \mathbb{N}$ gives a set of all acceptable binary strings of DFA. We denote this DFA by $x \equiv 0(\text{mod } n)$.

Example 3.1. Construct a DFA with remainder 2 accepts the set of all binary strings that interpreted as the binary representation of an unsigned decimal integer, is divisible by 3.

Since, when a number is divided by 3, the possible remainders are $0,1,2$. So, we can assign each state as q_0, q_1, q_2. For this, consider $\sum = (Q, A, \delta, q_0^*, F)$, where $Q = \{q_0, q_1, q_2\}$, $A = \{0,1\}, q_0^* = \{q_0\}$, $F = \{q_2\}$ and state transition Table 2 and the state transition Diagram 2.

Table 2. State transition table of $\sum$

δ Input → / States ↓	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

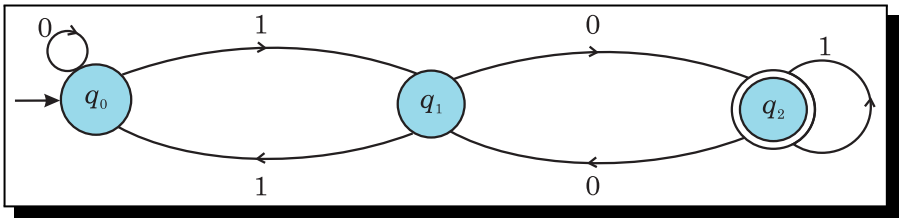

Diagram 2. A DFA of remainder 2 when it is divisible by 3

Here binary acceptable strings in this DFA are $\{10,101,1000,1011,1110,\ldots\}$ which represents $\{2,5,8,11,14,\ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 2(\text{mod } 3)$ in decimal systems.

3.1 Automata in Chinese Remainder Theorem

CRT is one of the pearl of number theory. The CRT is a theorem which gives a unique solution to simultaneous linear congruences with co-prime moduli. Now, we discuss some problems based on CRT by using finite automata with recent concepts.

Example 3.2. Solve the linear congruences $x \equiv 1(\text{mod } 2)$, $x \equiv 2(\text{mod } 3)$.

First, we solve the problem by using CRT, here $a_1 = 1$, $a_2 = 2$, $m_1 = 2$, $m_2 = 3$, $M = m_1.m_2 = 2.3 = 6$, $M_1 = \frac{M}{m_1} = 3$, $M_2 = \frac{M}{m_2} = 2$. We have to find solutions for $3y_1 \equiv 1(\text{mod } 2) \Rightarrow y_1 \equiv 1(\text{mod } 2)$ and $2y_2 \equiv 1(\text{mod } 3)$, by inspection, $y_1 = 1$, $y_2 = 2$. Therefore, $x \equiv (a_1M_1y_1 + a_2M_2y_2)(\text{mod } M) \equiv (1.3.1 + 2.2.2)(\text{mod } 6) \equiv 5(\text{mod } 6)$.

Now, solve the problem by using automata theory. Let us consider $\sum_1 = (Q_1, A_1, q_0', \delta_1, F_1)$, $q_0' \in Q_1$, $F_1 \subseteq Q_1$ with $Q_1 = \{p_0, p_1\}$, $A_1 = \{0,1\}$, $F_1 = \{p_1\}$, $\delta_1 : Q_1 \times A_1 \to Q_1$ defined by the state transition Table 3 and the state transition Diagram 3.

Table 3. State transition table of $\sum_1$

δ_1 Input → / States ↓	0	1
p_0	p_0	p_1
p_1	p_0	p_1

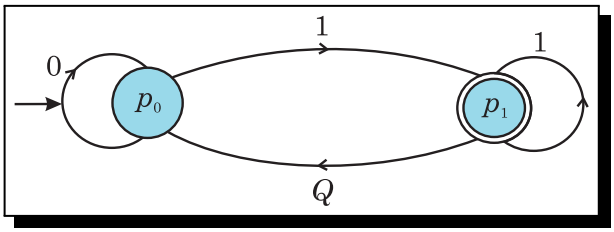

Diagram 3. A DFA of remainder 1 when it is divisible by 2

Here binary acceptable strings in this DFA are $\{1,11,101,111,1001,1011,\ldots\}$ which represents $\{1,3,5,7,9,11,\ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 1(\text{mod } 2)$ in decimal systems.

Again, consider another automaton $\sum_2 = (Q_2, A_2, q_0'', \delta_2, F_2)$, $q_0'' \in Q_2$, $F_2 \subseteq Q_2$ with $Q_2 = \{q_0, q_1, q_2\}$, $q_0'' = \{q_0\}$, $A_2 = \{0,1\}$, $F_2 = \{q_2\}$, $\delta_2 : Q_2 \times A_2 \to Q_2$ defined by the state transition Table 4 and the state transition Diagram 2.

Table 4. State Transition table of $\sum_2$

δ_2 Input → / States ↓	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

Here binary acceptable strings in this DFA are $\{10,101,1000,1011,1110,\ldots\}$ which represents $\{2,5,8,11,14,\ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 2(\text{mod } 3)$ in decimal systems.

Now, used the definition of Cartesian product of finite automata on $\sum_1$ and $\sum_2$. Therefore, $\sum = \sum_1 \times \sum_2$ is given by $\sum = (Q, A, \delta, q_0^*, F)$ where $Q = Q_1 \times Q_2 = \{(p_0,q_0),(p_0,q_1),(p_0,q_2),(p_1,q_0),(p_1,q_1),(p_1,q_2)\}$, $A = A_1 = A_2 = \{0,1\}$, $q_0^* = \{(p_0,q_0)\}$, $F = \{(p_1,q_2)\}$, $\delta = \delta_1 \times \delta_2$ with transition function $\delta : (Q_1 \times Q_2) \times A \to Q_1 \times Q_2$ is defined by the transition Table 5 and the state transition Diagram 4.

Table 5. State Transition table of $\sum$

δ Input → / States ↓	0	1
(p_0,q_0)	(p_0,q_0)	(p_1,q_1)
(p_0,q_1)	(p_0,q_2)	(p_1,q_0)
(p_0,q_2)	(p_0,q_1)	(p_1,q_2)
(p_1,q_0)	(p_0,q_0)	(p_1,q_1)
(p_1,q_1)	(p_0,q_2)	(p_1,q_0)
(p_1,q_2)	(p_0,q_1)	(p_1,q_2)

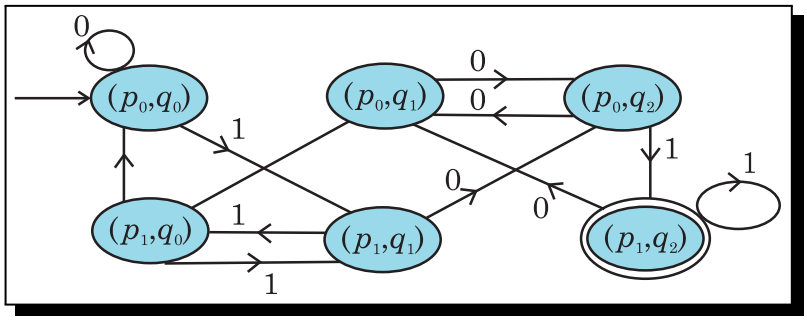

Diagram 4. A DFA of remainder 5 when it is divisible by 6

Here binary acceptable strings in this DFA are $\{101,1011,10001,10111,11101,\ldots\}$ which represents $\{5,11,17,23,29,\ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 5(\text{mod } 6)$ in decimal systems.

Example 3.3. Solve the linear congruences $x \equiv 2(\text{mod } 3)$, $x \equiv 3(\text{mod } 5)$.

First, we solve the problem by using CRT, here $a_1 = 2$, $a_2 = 3$, $m_1 = 3$, $m_2 = 5$, $M = m_1.m_2 = 3.5 = 15$, $M_1 = \frac{M}{m_1} = 5$, $M_2 = \frac{M}{m_2} = 3$. We have to find solutions for $5y_1 \equiv 1(\text{mod } 3) \Rightarrow 2y_1 \equiv 1(\text{mod } 3)$ and $3y_2 \equiv 1(\text{mod } 5)$, by inspection, $y_1 = 2, y_2 = 2$. Therefore, $x \equiv (a_1M_1y_1 + a_2M_2y_2)(\text{mod } M) \equiv (2.5.2 + 3.3.2)(\text{mod } 15) \equiv 8(\text{mod } 15)$.

Now, solve the problem by using automata theory. Let us consider $\sum_1 = (Q_1, A_1, q_0', \delta_1, F_1)$, $q_0' \in Q_1$, $F_1 \subseteq Q_1$ with $Q_1 = \{q_0, q_1, q_2\}$, $A_1 = \{0,1\}$, $q_0' = \{q_0\}$, $F_1 = \{q_2\}$, $\delta_1 : Q_1 \times A_1 \to Q_1$ defined by the state transition Table 6 and the state transition Diagram 2.

Table 6. State Transition table of $\sum_1$

δ_1 Input → / States ↓	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

Here binary acceptable strings in this DFA are $\{10,101,1000,1011,1110,\ldots\}$ which represents $\{2,5,8,11,14,\ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 2(\text{mod } 3)$ in decimal systems.

Again, consider another automaton $\sum_2 = (Q_2, A_2, q_0'', \delta_2, F_2)$, $q_0'' \in Q_2$, $F_2 \subseteq Q_2$ with $Q_2 = \{r_0, r_1, r_2, r_3, r_4\}$, $q_0'' = \{r_0\}$, $A_2 = \{0,1\}$, $F_2 = \{r_3\}$, $\delta_2 : Q_2 \times A_2 \to Q_2$ defined by the state transition Table 7 and the state transition Diagram 5.

Table 7. State Transition table of $\sum_2$

δ_2 Input → / States ↓	0	1
r_0	r_0	r_1
r_1	r_2	r_3
r_2	r_4	r_0
r_3	r_1	r_2
r_4	r_3	r_4

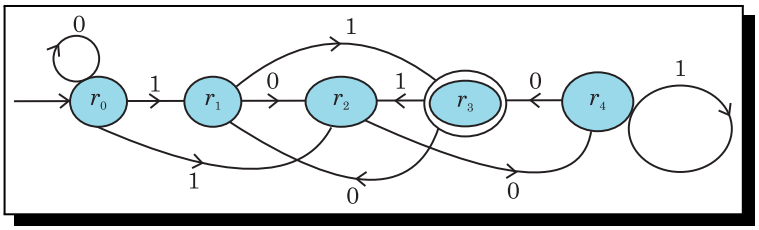

Diagram 5. A DFA of remainder 3 when it is divisible by 5

Here binary acceptable strings in this DFA are $\{11, 1000, 1101, 10010, 10111, \ldots\}$ which represents $\{3, 8, 13, 18, 23, \ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 3 (\text{mod } 5)$ in decimal systems. So $\sum_2$ gives the linear congruence $x \equiv 3 (\text{mod } 5)$.

Table 8. State Transition table of $\sum$

δ Input → / States ↓	0	1
(q_0, r_0)	(q_0, r_0)	(q_1, r_1)
(q_0, r_1)	(q_0, r_2)	(q_1, r_3)
(q_0, r_2)	(q_0, r_4)	(q_1, r_0)
(q_0, r_3)	(q_0, r_1)	(q_1, r_2)
(q_0, r_4)	(q_0, r_3)	(q_1, r_4)
(q_1, r_0)	(q_2, r_0)	(q_0, r_1)
(q_1, r_1)	(q_2, r_2)	(q_0, r_3)
(q_1, r_2)	(q_2, r_4)	(q_1, r_0)
(q_1, r_3)	(q_2, r_1)	(q_1, r_2)
(q_1, r_4)	(q_2, r_3)	(q_1, r_4)
(q_2, r_0)	(q_1, r_0)	(q_2, r_1)
(q_2, r_1)	(q_1, r_2)	(q_2, r_3)
(q_2, r_2)	(q_1, r_4)	(q_2, r_0)
(q_2, r_3)	(q_1, r_1)	(q_2, r_2)
(q_2, r_4)	(q_1, r_3)	(q_2, r_4)

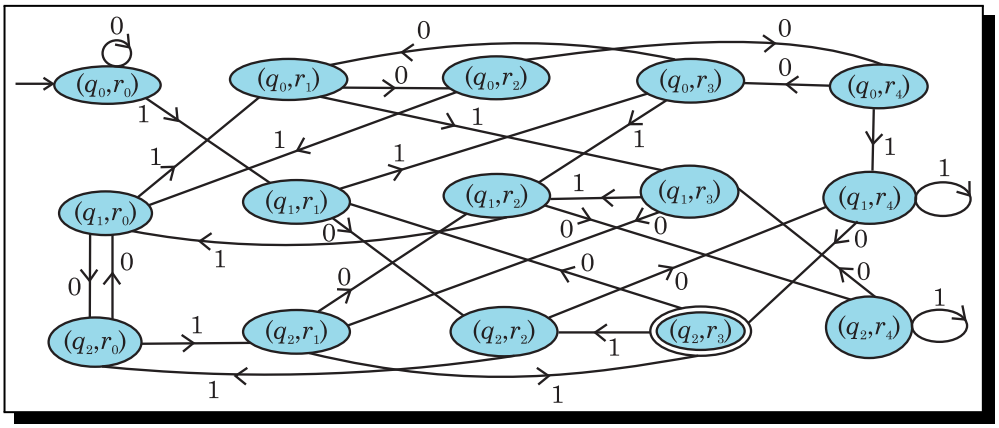

Diagram 6. A DFA of remainder 8 when it is divisible by 15

Now, used the definition of Cartesian product of finite automata on $\sum_1$ and $\sum_2$. Therefore, $\sum = \sum_1 \times \sum_2$ is given by $\sum = (Q, A, \delta, q_0^*, F)$ where $Q = Q_1 \times Q_2 = \{(q_0,r_0), (q_0,r_1), (q_0,r_2), (q_0,r_3), (q_0,r_4), (q_1,r_0), (q_1,r_1), (q_1,r_2), (q_1,r_3), (q_1,r_4), (q_2,r_0), (q_2,r_1), (q_2,r_2), (q_2,r_3), (q_2,r_4)\}$, $A = A_1 = A_2 = \{0,1\}$, $q_0^* = \{(q_0,r_0)\}$, $F = \{(q_2,r_3)\}$, $\delta = \delta_1 \times \delta_2$ with transition function $\delta : (Q_1 \times Q_2) \times A \to Q_1 \times Q_2$ is defined by the transition Table 8 and the state transition Diagram 6.

Here binary acceptable strings in this DFA are $\{1000, 10111, 100110, 110101 \ldots\}$ which represents $\{8, 23, 38, 53, \ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 8 (\text{mod } 15)$ in decimal systems.

Example 3.4. Solve the linear congruences $x \equiv 1(\text{mod } 2)$, $x \equiv 2(\text{mod } 3)$, $x \equiv 3(\text{mod } 5)$.

First, solve the problem by using CRT, here $a_1 = 1$, $a_2 = 2$, $a_3 = 3m_1 = 2$, $m_2 = 3$, $m_3 = 5$, $M = m_1.m_2.m_3 = 2.3.5 = 30$, $M_1 = \frac{M}{m_1} = 15$, $M_2 = \frac{M}{m_2} = 10$, $M_3 = \frac{M}{m_3} = 6$. We have to find solutions for $15y_1 \equiv 1(\text{mod } 2) \Rightarrow y_1 \equiv 1(\text{mod } 2)$, $10y_2 \equiv 1(\text{mod } 3) \Rightarrow y_2 \equiv 1(\text{mod } 3)$, $6y_3 \equiv 1(\text{mod } 5) \Rightarrow y_3 \equiv 1(\text{mod } 5)$, by inspection, $y_1 = 1$, $y_2 = 1$, $y_3 = 1$. Therefore, $x \equiv (a_1M_1y_1 + a_2M_2y_2 + a_3M_3y_3)(\text{mod } M) \equiv (1.15.1 + 2.10.1 + 3.6.1)(\text{mod } 30) \equiv 23(\text{mod } 30)$.

Now, solve the problem by using automata theory. Consider $\sum_1 = (Q_1, A_1, q_0', \delta_1, F_1)$, $q_0' \in Q_1$, $F_1 \subseteq Q_1$ with $Q_1 = \{p_0, p_1\}$, $A_1 = \{0,1\}$, $q_0' = \{p_0\}$, $F_1 = \{p_1\}$, $\delta_1 : Q_1 \times A_1 \to Q_1$ defined by the state transition Table 3 and state transition Diagram 3.

Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 1(\text{mod } 2)$ in decimal systems.

Consider another automaton $\sum_2 = (Q_2, A_2, q_0'', \delta_2, F_2)$, $q_0'' \in Q_2$, $F_2 \subseteq Q_2$ with $Q_2 = \{q_0, q_1, q_2\}$, $A_1 = \{0,1\}$, $q_0'' = \{q_0\}$, $F_2 = \{q_2\}$, $\delta_2 : Q_2 \times A_2 \to Q_2$ defined by the state transition Table 4 and state transition Diagram 2.

Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 2(\text{mod } 3)$ in decimal systems.

Again, consider another automaton $\sum_3 = (Q_3, A_3, q_0''', \delta_3, F_3)$, $q_0''' \in Q_3$, $F_3 \subseteq Q_3$ with $Q_3 = \{r_0, r_1, r_2, r_3, r_4\}$, $A_1 = \{0,1\}$, $q_0''' = \{r_0\}$, $F_3 = \{r_3\}$, $\delta_3 : Q_3 \times A_3 \to Q_3$ defined by the state transition Table 9 and the state transition Diagram 5.

Table 9. State Transition table of $\sum_3$

States ↓ / δ_3 Input →	0	1
r_0	r_0	r_1
r_1	r_2	r_3
r_2	r_4	r_0
r_3	r_1	r_2
r_4	r_3	r_4

Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 3(\text{mod } 5)$ in decimal systems.

Table 10. State Transition table of $\sum$

States ↓ / δ Input →	0	1
(p_0,q_0,r_0)	(p_0,q_0,r_0)	(p_1,q_1,r_1)
(p_0,q_0,r_1)	(p_0,q_0,r_2)	(p_1,q_1,r_3)
(p_0,q_0,r_2)	(p_0,q_0,r_4)	(p_1,q_1,r_0)
(p_0,q_0,r_3)	(p_0,q_0,r_1)	(p_1,q_1,r_2)
(p_0,q_0,r_4)	(p_0,q_0,r_3)	(p_1,q_1,r_4)
(p_0,q_1,r_0)	(p_0,q_2,r_0)	(p_1,q_0,r_1)
(p_0,q_1,r_1)	(p_0,q_2,r_2)	(p_1,q_0,r_3)
(p_0,q_1,r_2)	(p_0,q_2,r_4)	(p_1,q_0,r_0)
(p_0,q_1,r_3)	(p_0,q_2,r_1)	(p_1,q_0,r_2)
(p_0,q_1,r_4)	(p_0,q_2,r_3)	(p_1,q_0,r_4)
(p_0,q_2,r_0)	(p_0,q_1,r_0)	(p_1,q_2,r_1)
(p_0,q_2,r_1)	(p_0,q_1,r_2)	(p_1,q_2,r_3)
(p_0,q_2,r_2)	(p_0,q_1,r_4)	(p_1,q_2,r_0)
(p_0,q_2,r_3)	(p_0,q_1,r_1)	(p_1,q_2,r_2)
(p_0,q_2,r_4)	(p_0,q_1,r_3)	(p_1,q_2,r_4)
(p_1,q_0,r_0)	(p_0,q_0,r_0)	(p_1,q_1,r_1)
(p_1,q_0,r_1)	(p_0,q_0,r_2)	(p_1,q_1,r_3)
(p_1,q_0,r_2)	(p_0,q_0,r_4)	(p_1,q_1,r_0)
(p_1,q_0,r_3)	(p_0,q_0,r_1)	(p_1,q_1,r_2)
(p_1,q_0,r_4)	(p_0,q_0,r_3)	(p_1,q_1,r_4)
(p_1,q_1,r_0)	(p_0,q_2,r_0)	(p_1,q_0,r_1)
(p_1,q_1,r_1)	(p_0,q_2,r_2)	(p_1,q_0,r_3)
(p_1,q_1,r_2)	(p_0,q_2,r_4)	(p_1,q_0,r_0)
(p_1,q_1,r_3)	(p_0,q_2,r_1)	(p_1,q_0,r_2)
(p_1,q_1,r_4)	(p_0,q_2,r_3)	(p_1,q_0,r_4)
(p_1,q_2,r_0)	(p_0,q_1,r_0)	(p_1,q_2,r_1)
(p_1,q_2,r_1)	(p_0,q_1,r_2)	(p_1,q_2,r_3)
(p_1,q_2,r_2)	(p_0,q_1,r_4)	(p_1,q_2,r_0)
(p_1,q_2,r_3)	(p_0,q_1,r_1)	(p_1,q_2,r_2)
(p_1,q_2,r_4)	(p_0,q_1,r_3)	(p_1,q_2,r_4)

Now, used the definition of Cartesian product of finite automata on $\sum_1$, $\sum_2$ and $\sum_3$. Therefore, $\sum = \sum_1 \times \sum_2 \times \sum_3$ is given by $\sum = (Q, A, \delta, q_0^*, F)$ where $Q = Q_1 \times Q_2 \times Q_3 = \{(p_0,q_0,r_0), (p_0,q_0,r_1), (p_0,q_0,r_2), (p_0,q_0,r_3), (p_0,q_0,r_4), (p_0,q_1,r_0), (p_0,q_1,r_1), (p_0,q_1,r_2), (p_0,q_1,r_3), (p_0,q_1,r_4), (p_0,q_2,r_0), (p_0,q_2,r_1), (p_0,q_2,r_2), (p_0,q_2,r_3), (p_0,q_2,r_4), (p_1,q_0,r_0), (p_1,q_0,r_1), (p_1,q_0,r_2), (p_1,q_0,r_3), (p_1,q_0,r_4), (p_1,q_1,r_0), (p_1,q_1,r_1), (p_1,q_1,r_2), (p_1,q_1,r_3), (p_1,q_1,r_4), (p_1,q_2,r_0), (p_1,q_2,r_1), (p_1,q_2,r_2), (p_1,q_2,r_3), (p_1,q_2,r_4)\}$, $A = A_1 = A_2 = \{0,1\}$, $q_0^* = \{(p_0,q_0,r_0)\}$, $F = \{(p_1,q_2,r_3)\}$, $\delta = \delta_1 \times \delta_2 \times \delta_3$ with transition function $\delta : (Q_1 \times Q_2 \times Q_3) \times A \to Q_1 \times Q_2 \times Q_3$ is defined by the transition Table 10.

Here we assign state (p_0,q_0,r_0) as 1, state (p_0,q_0,r_1) as 2, state (p_0,q_0,r_2) as 3 and so on. The state transition diagram $\sum$ is shown in Diagram 7.

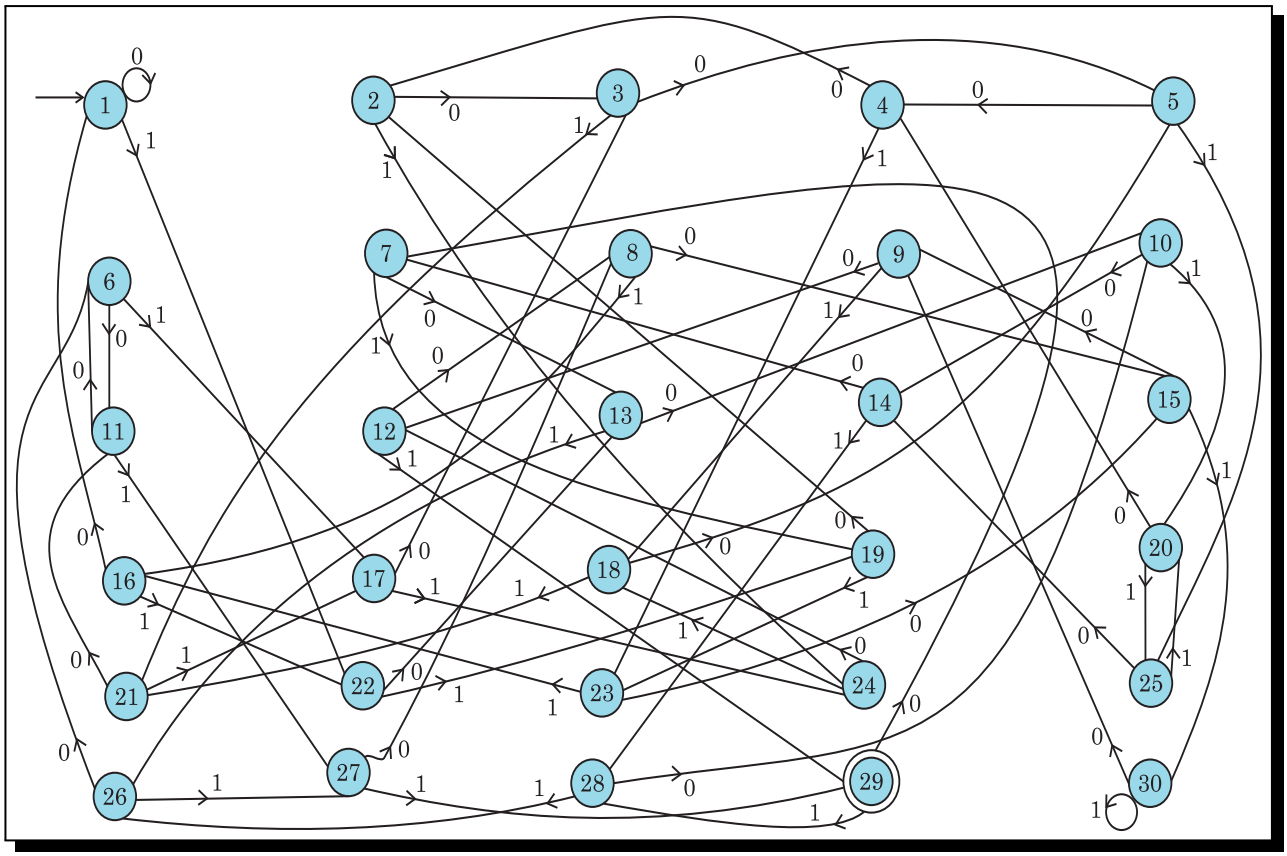

Diagram 7. A DFA of remainder 23 when it is divisible by 30

Here binary acceptable strings in this DFA are $\{10111, 110101, 1010011\ldots\}$ which represents $\{23, 53, 83, \ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 23 (\text{mod } 30)$ in decimal systems.

4. Observations

(a) The congruences $x \equiv a (\text{mod } m)$, $x \equiv b (\text{mod } n)$ has no solution if $(m,n) \neq 1$, $m,n,a,b \in \mathbb{Z}$. For this, we consider two congruences $x \equiv 1 (\text{mod } 2)$, $x \equiv 0 (\text{mod } 2)$ which has no solution. Now,

solve the problem by using automata theory. Let us consider $\sum_1 = (Q_1, A_1, q_0', \delta_1, F_1)$, $q_0' \in Q_1$, $F_1 \subseteq Q_1$ with $Q_1 = \{p_0, p_1\}$, $A_1 = \{0,1\}$, $q_0' = \{p_o\}$, $F_1 = \{p_1\}$, $\delta_1 : Q_1 \times A_1 \to Q_1$ defined by the state transition Table 3 and state transition Diagram 3. Here binary acceptable strings in this DFA are $\{1, 11, 101, 111, 1001, 1011, \ldots\}$ which represents $\{1, 3, 5, 7, 9, 11, \ldots\}$ in decimal system. Therefore, we see in the diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 1 (\text{mod } 2)$ in decimal systems. Again, Consider another automaton $\sum_2 = (Q_2, A_2, q_0'', \delta_2, F_2)$, $q_0'' \in Q_2$, $F_2 \subseteq Q_2$ with $Q_2 = \{q_0, q_1\}$, $q_0'' = \{q_0\}$, $A_2 = \{0,1\}$, $F_2 = \{q_0\}$, $\delta_2 : Q_2 \times A_2 \to Q_2$ defined by the state transition Table 11 and the state transition diagram is in Figure 8.

Table 11. State Transition table of $\sum_2$

δ_2 Input → / States ↓	0	1
q_0	q_0	q_1
q_1	q_0	q_1

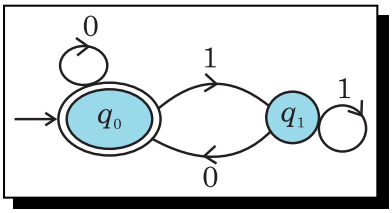

Diagram 8. A DFA of remainder 0 when it is divisible by 2

Here binary acceptable strings in this DFA are $\{0, 10, 100, 110, 1000, \ldots\}$ which represents $\{0, 2, 4, 6, 8, \ldots\}$ in decimal system. Therefore, we see in the above diagram that the collection of those acceptable binary strings which gives the final state of the automata represents the solutions of the linear congruence $x \equiv 0 (\text{mod } 2)$ in decimal systems.

Now, used the definition of Cartesian product of finite automata on $\sum_1$ and $\sum_2$. Therefore, $\sum = \sum_1 \times \sum_2$ is given by $\sum = (Q, A, \delta, q_0^*, F)$ where $Q = Q_1 \times Q_2 = \{(p_0, q_0), (p_0, q_1), (p_1, q_0), (p_1, q_1)\}$, $A = A_1 = A_2 = \{0,1\}$, $q_0^* = \{(p_0, q_0)\}$, $F = \{(p_1, q_0)\}$, $\delta = \delta_1 \times \delta_2$ with transition function $\delta : (Q_1 \times Q_2) \times A \to Q_1 \times Q_2$ is defined by the transition Table 12 and the state transition Diagram 9.

Table 12. State Transition table of $\sum$

δ Input → / States ↓	0	1
(p_0, q_0)	(p_0, q_0)	(p_1, q_1)
(p_0, q_1)	(p_0, q_0)	(p_1, q_1)
(p_1, q_0)	(p_0, q_0)	(p_1, q_1)
(p_1, q_1)	(p_0, q_0)	(p_1, q_1)

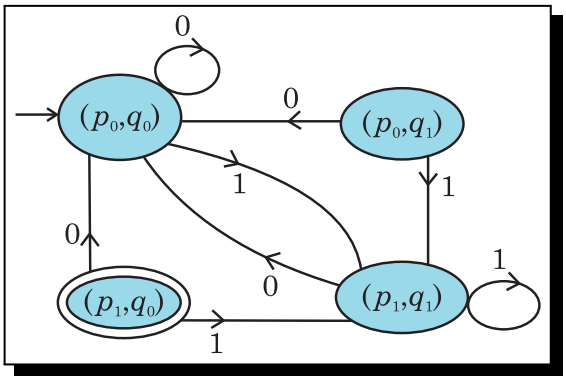

Diagram 9. A DFA of No Solutions

Therefore, we see in the above diagram of the DFA that there is no acceptable binary strings which provide the final state of the automata represent that there is no solution of the listed linear congruences.

(b) If this technique can be extended for k congruences of Chinese Remainder Theorem, k-automata like $\sum_1, \sum_2, \sum_3, \ldots, \sum_k$ can be found. Then, use the definition of Cartesian product of finite automata on k-automata, we get another automaton $\sum = \sum_1 \times \sum_2 \times \sum_3 \times \ldots \times \sum_k$ is given by $\sum = (Q, A, \delta, q_0^*, F)$ where $Q = Q_1 \times Q_2 \times Q_3 \times \ldots \times Q_k$ which contains $m_1 \cdot m_2 \cdot m_3 \cdot \ldots \cdot m_k$ elements, $A = A_1 = A_2 = A_3 = \ldots = A_k = \{0,1\}$, $q_0^* = \{(p_1, p_2, p_3, \ldots p_k)\}$, $\{p_1\} \in Q_1$, $\{p_2\} \in Q_2, \ldots \{p_k\} \in Q_k$, $(q_1, q_2, \ldots q_k) \in F$ iff $q_1 \in F_1, q_2 \in F_2, \ldots q_k \in F_k$, $\delta = \delta_1 \times \delta_2 \times \ldots \times \delta_k$ with state transition function is $\delta : (Q_1 \times Q_2 \times \ldots \times Q_k) \times A \to Q_1 \times Q_2 \times \ldots \times Q_k$ defined by $\delta((q_1, q_2, \ldots q_k), a) = (\delta_1(q_1, a), \delta_2(q_2, a), \ldots \delta_k(q_k, a))$, $q_1 \in Q_1$, $q_2 \in Q_2, \ldots q_k \in Q_k$, $a \in A$. This $\sum$ also gives us a state transition diagram with an acceptable infinite set of binary strings which gives the final state of the automaton represents the solutions of the linear congruence against $\sum$ in decimal systems.

5. Conclusion

Finite automata are a branch of process design concerned with string manipulation and sequence processing. We concentrate on how specific problems in which the Chinese Remainder Theorem can be attacked in a novel approach using a technique from the theory of finite automata. The main result of the paper is that residue classes can be recognized by finite automaton. The result can be expanded to exploration into other results of the Chinese Remainder Theorem as well as number theory by utilizing automata theory.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] B. Adamczewski and J. Bell, *Automata in Number Theory* (Chapter 25), CNRS and University of Waterloo (2018), URL: https://adamczewski.perso.math.cnrs.fr/Chapter25.pdf.

[2] J.P. Allouche, Cellular automata, finite automata, and number theory, in: *Cellular Automata*, pp. 321 – 330, (1999), *Mathematics and Its Applications book series* (MAIA, Vol. 460), Springer, Dordrecht, DOI: 10.1007/978-94-015-9153-9_13.

[3] D.M. Burton, *Elementary Number Theory*, Tata McGraw-Hill Education (2006).

[4] C. Ding, D. Pei and A. Salomaa, *Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography*, World Scientific Publishing Co. Pte. Ltd., Singapore (1996), DOI: 10.1142/3254.

[5] W. Dörfler, The cartesian product of automata, *Mathematical System Theory* **11** (1978), 239 – 257, DOI: 10.1007/BF01768479.

[6] V.M. Glushkov, The abstract theory of automata, *Russian Mathematical Surveys (Uspekhi Matematicheskikh Nauk)* **16**(5) (1961), 3 – 62, URL: http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=rm&paperid=6668&option_lang=eng (in Russian).

[7] J. Hartmanis and H. Shank, On the recognition of primes by automata, *Journal of the ACM* **15**(3) (1968), 382 – 389, DOI: 10.1145/321466.321470.

[8] W.M.L. Holcombe, *Algebraic Automata Theory*, in: Cambridge Studies in Mathematics series, Vol. 1, Cambridge University Press (1981), URL: https://www.ioc.ee/~jaan/Algebraic_automata/Holcombe/Holcombe1.PDF

[9] J.E. Hopcroft, R. Motwani and J.D. Ullman, Introduction to automata theory, languages, and computation, *ACM SIGACT News* **32**(1) (2001), 60 – 65, DOI: 10.1145/568438.568455.

[10] S.C. Hsieh, Product construction of finite-state machines, in: *Proceedings of the World Congress on Engineering and Computer Science, Vol. I* (WCECS 2010), October 20-22, 2010, San Francisco, USA, pp. 141 – 143, (2010), http://www.iaeng.org/publication/WCECS2010/WCECS2010_pp141-143.pdf.

[11] S. Kandar, *Introduction to Automata Theory, Formal Languages and Computation*, Pearson Education India (2013).

[12] K.L.P. Mishra and N. Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI Learning Pvt. Ltd. (2006).

[13] D.E. Muller, Theory of automata, in: F. Preparata (eds.), *Theoretical Computer Science. C.I.M.E. Summer Schools*, Vol. 68, Springer, Berlin — Heidelberg, (2011), DOI: 10.1007/978-3-642-11120-4_2.

[14] D. Perrin, Formal models and semantics, in: *Handbook of Theoretical Computer Science*, J. van Leeuwen (ed.), 3 – 57 (1990), DOI: 10.1016/B978-0-444-88074-1.50006-8.

[15] J.E. Pin, F*ormal Properties of Finite Automata and Applications*, in: Proceedings of LITP Spring School on Theoretical Computer Science, Ramatuelle, France, May 23-27, 1988, Springer Science & Business Media (1989), https://link.springer.com/book/10.1007/BFb0013106.

[16] A. Rajasekaran, J. Shallit and T. Smith, Additive number theory via automata theory, *Theory of Computing Systems* **64** (2020), 542 – 567, DOI: 10.1007/s00224-019-09929-9.

[17] G. Rauzy, Numbers and automata, in: *LITP 1988: Formal Properties of Finite Automata and Applications*, J.E. Pin (ed.), *Lecture Notes in Computer Science book series* (LNCS, Vol. 386), Springer, Berlin — Heidelberg, 176 – 185 (1988), DOI: 10.1007/BFb0013120.

[18] A. Restivo, Codes and automata, in: *LITP 1988: Formal Properties of Finite Automata and Applications*, J.E. Pin (ed.), *Lecture Notes in Computer Science book series* (LNCS, Vol. 386), Springer, Berlin — Heidelberg, 186 – 198 (1988), DOI: 10.1007/BFb0013121.

[19] M. Rigo, Formal languages, automata and numeration systems, in: *Applications to Recognizability and Decidability*, Vol. 2, John Wiley & Sons (2014), https://orbi.uliege.be/handle/2268/176351.

[20] A. Salomaa, *Theory of Automata*, 1st edition, Elsevier (1969), URL: https://www.elsevier.com/books/theory-of-automata/salomaa/978-0-08-013376-8.

[21] W. Steiner, *Numeration Systems: Automata, Combinatorics, Dynamical Systems, Number Theory*, Doctoral dissertation, Institut de Recherche en Informatique Fondamenta, Université de Paris (2021), URL: https://www.irif.fr/~steiner/hdr.pdf.

[22] S. Wolfram, *Cellular Automata and Complexity*, 1st edition, CRC Press, Boca Raton (2019), DOI: 10.1201/9780429494093.