

ALGORITHMIC SOLVABILITY OF THE ISOMORPHISM PROBLEM FOR AUTOMATA

1961

SovietRxiv

View the original and related papers at <https://sovietrxiv.org/items/ru-196101.90515>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

Abstract

Full Text

CYBERNETICS AND CONTROL THEORY

Yu. I. SORKIN

ALGORITHMIC SOLVABILITY OF THE ISOMORPHISM PROBLEM FOR AUTOMATA

(Presented by Academician P. S. Novikov on 2 XI 1960)

Let an arbitrary set Λ be given, whose elements we shall call the internal states, or, more briefly, the states of a partial automaton. Let a second set \mathfrak{A} be given, whose elements we shall call inputs. It is assumed here that the set of inputs \mathfrak{A} is nonempty and does not intersect the set of states Λ . Let to each input $a \in \mathfrak{A}$ there be assigned some subset Λ_a (of Λ), called the support of the input a . Finally, we shall suppose that each input a from \mathfrak{A} is a mapping of its support Λ_a into the set of all states Λ , i.e. $\lambda a \in \Lambda$ for any $a \in \mathfrak{A}$ and $\lambda \in \Lambda_a$.

Under these assumptions, by a partial automaton $\Lambda(\mathfrak{A})$ we shall mean the set Λ together with a specified set \mathfrak{A} of mappings of subsets Λ_a of Λ into the set Λ . If in the partial automaton $\Lambda(\mathfrak{A})$ the supports of all inputs coincide with Λ , i.e. $\Lambda_a = \Lambda$ for every $a \in \mathfrak{A}$, then such a partial automaton will be called a complete automaton, or, more briefly, an automaton. In the case where the sets Λ and \mathfrak{A} are finite, the definition of an automaton just formulated coincides with the known definition of the Moore-Yu. T. Medvedev machine-automaton ⁽¹⁾–⁽²⁾.

It is easy to see that automata and partial automata are a special case of general algebraic systems and, respectively, partial algebraic systems ⁽³⁾. Therefore for partial automata such notions of the general theory of algebraic systems as isomorphism, homomorphism, automorphism, subautomaton, system of generators, free automaton, etc., are quite meaningful. In particular, for automata the theory of defining them by defining relations is constructed in a natural way.

For automata, theorems analogous to the group-theoretic ⁽⁴⁾ theorems of Nielsen-Schreier, Dick, Tietze, etc., are easily proved. The theorem on the embedding of countable automata, i.e. automata in which the set $\Lambda \cup \mathfrak{A}$ has countable cardinality, into automata with a finite number of generators is true.

Theorem A. *Every countable automaton $\Lambda(\mathfrak{A})$ with at least one input is isomorphic to a subautomaton of some automaton with one generating state and with the same set of inputs \mathfrak{A} .*

In connection with the specification of automata by defining relations for them, as for other algebraic systems, there arises a series of mass algorithmic problems analogous to those first posed by Dehn ⁽⁵⁾ in group theory, namely the problems:

1) of word identity, 2) of membership of words in a subautomaton, and 3) of isomorphism. For these three mass problems, as well as for some others in the theory of automata, one can indicate simple decision algorithms. Before proceeding to the description of these algorithms and to the formulation of the theorems justifying them, we shall clarify the statement of the problems.

Let two arbitrary automata $\Lambda(\mathfrak{A})$ and $\Lambda_1(\mathfrak{A})$ be given by finite systems of generating states, inputs, and finite systems of defining relations. The identity problem: to find an algorithm that determines—

for arbitrary two words A and B from the automaton $\Lambda(\mathfrak{A})$, whether they are equal or not, i.e. $A = B$ or $A \neq B$? The membership problem: to find an algorithm which determines, for arbitrary $m + 1$ words A, B_1, B_2, \dots, B_m from the automaton $\Lambda(\mathfrak{A})$, whether the word A belongs to the subautomaton generated by the words B_1, B_2, \dots, B_m or not, i.e. $A \in \{B_1, B_2, \dots, B_m\}$ or $A \notin \{B_1, B_2, \dots, B_m\}$? The isomorphism problem: to find an algorithm which determines whether the automata $\Lambda(\mathfrak{A})$ and $\Lambda_1(\mathfrak{A})$ are isomorphic or not, i.e. $\Lambda(\mathfrak{A}) \cong \Lambda_1(\mathfrak{A})$ or $\Lambda(\mathfrak{A}) \not\cong \Lambda_1(\mathfrak{A})$?

The existence of an algorithm solving the identity problem for automata follows easily from Ivens' s theorem ⁽⁶⁾ and from the fact that every partial automaton $\Lambda'(\mathfrak{A})$ is embeddable in some (complete) automaton $\Lambda(\mathfrak{A})$. Such an embedding can be carried out, for example, by adding to Λ a new element τ and extending the results of the actions of inputs a from \mathfrak{A} on states λ' from $\Lambda' \cup \tau$, which were not defined in $\Lambda'(\mathfrak{A})$, by the formula $\lambda'a = \tau$. It follows from ⁽⁶⁾ that, for an arbitrary partial automaton $\Lambda'(\mathfrak{A})$, among the automata $\Lambda(\mathfrak{A})$ with the properties: $\Lambda(\mathfrak{A})$ contains a part $U(\mathfrak{A})$ such that $U(\mathfrak{A}) \cong \Lambda'(\mathfrak{A})$ and $\Lambda(\mathfrak{A})$ is generated by the set U , there exists a unique, up to isomorphism, automaton $\bar{\Lambda}(\mathfrak{A})$ such that every other automaton with the indicated property is a homomorphic image of the automaton $\bar{\Lambda}(\mathfrak{A})$. Such an automaton $\bar{\Lambda}(\mathfrak{A})$ is naturally called the **free extension** of the partial automaton $\Lambda'(\mathfrak{A})$.

Following ⁽⁶⁾, a system of defining relations R will be called **closed** if: 1) every defining relation from R has a left-hand side of the form $s_1 a$ and a right-hand side of the form s_2 , where s_1 and s_2 are generating states, and a is a generating input, and 2) in R there do not exist two defining relations with identical (graphically equal) left-hand sides. It follows from ⁽⁶⁾ that every (finitely defined) presentation of an automaton by defining relations can be **algorithmically** transformed into a presentation by a closed system of defining relations. Therefore in what follows one may assume that automata are given by closed systems of defining relations.

If an automaton $\Lambda(\mathfrak{A})$ is given by a closed system R of defining relations, then to every word $A \in \Lambda$ one can uniquely assign its **canonical representation** \bar{A} in the following way. If in the word A some subword A' coincides with the left-hand side of one of the defining relations from R , then we replace this subword by the right-hand side of the mentioned defining relation. Then the word A is transformed into a word A_1 . To the word A_1 the described transformation is

again applied, if it is applicable. It is clear that, by virtue of the closedness of the system of defining relations R , the described process, after a finite number of steps, uniquely leads to some word \bar{A} —the canonical representation of the word A .

Theorem . In an automaton $\Lambda(\mathfrak{A})$ given by a closed system of defining relations, arbitrary two words A and B are equal in the case when their canonical representations are graphically equal (identical): $\bar{A} \equiv \bar{B}$.

Theorem gives, first, an explicit algorithm for solving the identity problem and, second, an algorithm for solving the membership problem: $A \in \{B_1, B_2, \dots, B_m\}$ if and only if some word \bar{B}_i ($1 \leq i \leq m$) is graphically equal to some subword A_j of the word \bar{A} .

To justify an algorithm solving the isomorphism problem, we introduce the following notions. A state s of a partial automaton $\Lambda(\mathfrak{A})$ is called **removable** if and only if: 1) there exist $s_1 \in \Lambda$ and $a \in \mathfrak{A}$ such that $s = s_1 a$ and $s_1 \neq s$; 2) if $s_1 a_1 = s a_{i_1} a_{i_2} \dots a_{i_k} = s_2 a_2$, then $a_1 \equiv a_{i_k}$ and $s_1 = s a_{i_1} a_{i_2} \dots a_{i_{k-1}}$ for $k \geq 1$, while for $k = 0$ it is true that $s_1 = s_2$ and $a_1 = a_2$;

- 3) for an arbitrary sequence of inputs $a_{i_1}, a_{i_2}, \dots, a_{i_s}, a_{i_{s+1}}, a_{i_{s+2}}, \dots, a_{i_{s+k}}$ ($k \geq 1$), the inequality

$$s a_{i_1} a_{i_2} a_{i_3} \dots a_{i_s} a_{i_{s+1}} a_{i_{s+2}} \dots a_{i_{s+k}} \neq s a_{i_1} a_{i_2} \dots a_{i_s}$$

holds.

A state that is not removable is called **irremovable**. The set $K_{\Lambda(\mathfrak{A})}$ of all irremovable states of the partial automaton $\Lambda(\mathfrak{A})$ is called the **base** of the partial automaton $\Lambda(\mathfrak{A})$.

Theorem V. The base $K_{\Lambda(\mathfrak{A})}$ of any partial automaton $\Lambda(\mathfrak{A})$ with a finite number of generators and inputs is a system of generators.

Theorem G. Every automaton $\Lambda(\mathfrak{A})$ with a finite number of generating states and inputs is a free extension of its base.

Theorem D. If an automaton is given by a closed system of defining relations connecting the generating states s_1, s_2, \dots, s_n , then every word A whose canonical representation is distinct from s_i ($1 \leq i \leq n$) is a removable state.

It follows from Theorem D that if a definite automaton $\Lambda(\mathfrak{A})$ is given by a closed system of defining relations connecting the generating states s_1, s_2, \dots, s_n , then the base of the automaton is a subset of the set of generating states s_1, s_2, \dots, s_n .

Theorem E. If from a partial automaton $\Lambda(\mathfrak{A})$ one removes some set of removable states N and all states generated by them $\{N\}$, then in the remaining partial automaton $\Lambda \setminus \{N\}(\mathfrak{A})$ every state s will be removable or irremovable in exact accordance with whether it was removable or irremovable in the partial automaton $\Lambda(\mathfrak{A})$.

The results presented make it possible to indicate an algorithm solving the isomorphism problem for automata, consisting of the following stages:

- 1) The specifications of the two automata of interest to us are transformed into specifications by closed systems of defining relations.
- 2) For both automata the transition graphs Γ_1 and Γ_2 of the partial automata are drawn, consisting only of the generating states (connected by closed systems of defining relations).
- 3) For each state of Γ_1 and Γ_2 it is determined whether it is removable or not, and all removable states are deleted from Γ_1 and Γ_2 , i.e., only the bases K_1 and K_2 of both automata remain.
- 4) Having considered all possible mappings of the bases into each other, one determines whether they are isomorphic or not. If it turns out that $K_1 \cong K_2$, then for the two original automata we shall have the answer that they are isomorphic, since each of them is a free extension of one and the same base. If it turns out that $K_1 \not\cong K_2$, then for the original automata as well we shall have the answer that they are nonisomorphic, since otherwise we would have $K_1 \cong K_2$ by virtue of the uniqueness of the base in every partial automaton.

What has been established above permits us to assert the validity, for automorphisms of automata, of the following theorem:

Theorem Zh. Every finitely defined automaton has a finite automorphism group, whose Cayley table is algorithmically obtainable.

State All-Union
 Central Scientific Research
 Institute of Complex Automation

Received
 31 X 1960

REFERENCES

1. É. F. Moore, *Automata*, Moscow, 1956, p. 179.
2. Yu. T. Medvedev, *Automata*, Moscow, 1956, p. 385.
3. A. I. Mal'cev, *Matem. sborn.*, **35** (77), 3 (1954).
4. A. G. Kurosh, *Theory of Groups*, Moscow, 1953.
5. M. Dehn, *Math. Ann.*, **71**, 116 (1912).
6. T. Evans, *J. Lond. Math. Soc.*, **26**, 64 (1951).

Note: Figure translations are in progress. See original paper for figures.

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.