DOI 10.1007/s10559-016-9834-1
Cybernetics and Systems Analysis, Vol. 52, No. 3, May, 2016

CYBERNETICS

HARMONIZATION OF AUTOMATA SPECIFICATIONS REPRESENTED IN THE LANGUAGE L

A. N. Chebotarev UDC 519.713.1

Abstract. *Two methods are presented for the harmonization of automata specifications represented in the form of sets of clauses in the language L. Both methods are based on a technique of harmonization of automata that uses their parallel composition. Two ways of defining the semantics of the language L are described that are used in the harmonization methods.*

Keywords: *automaton specification, language L, harmonization of specifications, composition of specifications, composition correctness.*

INTRODUCTION

The automata-theoretic approach, i.e., the approach using automaton models is widely applied in designing reactive systems. However, the initial requirements on the functioning of a system can be much more conveniently specified in a logical language in the form of propositions that characterize necessary properties of the system being designed. The class of properties that can be formulated in a logical language determines its expressiveness. An extension of the expressiveness of a logical language is, as a rule, connected with increasing the computational complexity of design algorithms that use the language. Therefore, one has to search for a trade-off between the expressiveness of a language and the efficiency of methods for processing information represented in it. With a view to decreasing the complexity of such methods, the expressiveness of such a language called the language of initial specification in what follows is restricted. In this article, it is supposed that, for an initial specification, the language L [1] is used whose expressiveness is restricted to safety properties. These properties are representable by an X/Y-automaton (X/Y transducer), i.e., an automaton model widespread in the theory of design of reactive systems.

Since the passage from informal requirements to a formal specification of a system can be fraught with errors, it is necessary to have a tool checking (proving) some essential properties of such a specification. In particular, the consistency of a specification consisting of a large number of statements in a formal language is far from being obvious. Therefore, before the use of a specification for synthesizing automaton models, it is desirable to check it for consistency, which can turn out to be simpler than the synthesis of an empty automaton from its inconsistent specification. The interaction of the system being designed with its environment imposes, as a rule, additional constraints and, as a result, some independently taken consistent specification can turn out to be inconsistent with allowance for the interaction of the system being specified with its environment. The check for the possibility of such a situation is carried out by methods of harmonization of the specification of a system with the specification of its environment. This article that is a continuation the themes of [2] is devoted to the consideration of these methods. The methods proposed here are based on the methods offered in [2] for the harmonization of automaton models. The problem of harmonization of

V. M. Glushkov Institute of Cybernetics, National Academy of Sciences of Ukraine, Kyiv, Ukraine, *ancheb@gmail.com*. Translated from Kibernetika i Sistemnyi Analiz, No. 3, May–June, 2016, pp. 3–15. Original article submitted September 1, 2015.

specifications is formulated proceeding from the fact that the automaton synthesized from a specification harmonized with its environment must be harmonized with the automaton synthesized from the specification of its environment. The descriptions of the concepts of an X/Y-automaton, a Σ-automaton, a cyclic automaton, and a parallel composition of automata that are used in this article and also the references to works of foreign authors that pertain to this subject are given in [2].

LANGUAGE L: SYNTAX AND SEMANTICS

Before passing to the consideration of methods for the harmonization of specifications, we will give the exact description of the syntax of a specification in the language L, its representation in solving harmonization problems, and the automata semantics determining the class of equivalent automata described by the specification.

The language L [1] is a fragment of predicate logic with one-place predicates and a fixed interpretation domain represented by the set **Z** of integers (instants of time). A specification in the language L is of the form of a formula $\forall t f(t)$. Here, $f(t)$ is a formula constructed with the help of logical connectives from atomic formulas (atoms) of the form $p(t+k)$, where p is a one-place predicate symbol, t is a variable assuming values from the set of integers, and k is an integer called the rank of an atom. The difference between maximum and minimum values of ranks of the atoms occurring in a formula is called its depth. As is shown in [1], we can restrict ourselves to the consideration of only formulas $f(t)$ in which the maximum rank of atoms equals 0. In defining the automata semantics of specification languages, these languages and automata are considered as formalisms for the specification of sets of ω-words (infinite words) in an alphabet $\Sigma(\Omega)$, where $\Omega = \{p_1, \dots, p_m\}$ is a collection of predicate symbols (signature) of a specification. The symbols of the alphabet $\Sigma(\Omega)$ are binary vectors of length m.

Let Ω be the signature of a formula $f(t)$, and let r be its depth. We call a sequence $\langle \sigma_0, \sigma_1, \dots, \sigma_r \rangle$ of vectors from $\Sigma(\Omega)$ a state of depth r and the set $Q(r,\Omega)$ of all such sequences the space of states of depth r for the formula $f(t)$. We will sometimes consider the formula $f(t)$ as a propositional formula constructed from atoms $p_1(t), \dots, p_m(t)$, $p_1(t-1)$, $\dots$, $p_m(t-1), \dots, p_1(t-r), \dots, p_m(t-r)$. If components of a vector σ_i at a state $q = \langle \sigma_0, \sigma_1, \dots, \sigma_r \rangle$ are considered as truth values of the corresponding atoms of rank $i-r$ in the case of some ordering of the set Ω, then we can speak of the value of the formula $f(t)$ on the state q. Thus, we will consider that any propositional formula with these propositional variables specifies a subset of states of the space $Q(r,\Omega)$, namely, the states on which it is true. We assume that the formula $f(t)$ is true on a set of states $Q_1 \subseteq Q(r,\Omega)$ if it is true on at least one of these states and that it is false on the set Q_1 if it is false on all the states from Q_1.

On the set $Q(r,\Omega)$, we define the immediate successor relation N in such a manner that each state $q = \langle \sigma_0, \sigma_1, \dots, \sigma_r \rangle$ is immediately followed by $2^{|\Omega|}$ states of the form $\langle \sigma_1, \dots, \sigma_r, \sigma \rangle$, where $\sigma \in \Sigma(\Omega)$. The set of all states that immediately follow q is called the transition region for the state q and is denoted by $N(q)$. The immediate predecessor relation P is similarly defined in such a way that each state $q = \langle \sigma_0, \sigma_1, \dots, \sigma_r \rangle$ is immediately preceded by $2^{|\Omega|}$ states of the form $\langle \sigma, \sigma_1, \dots, \sigma_{r-1} \rangle$, where $\sigma \in \Sigma(\Omega)$.

In using the language L to specify X/Y-automata, predicate symbols are associated with input and output binary channels of an automaton, and the set of predicate symbols Ω is divided into two classes consisting of input and output symbols designated by U and W, respectively. The input alphabet of the Σ-automaton being specified is equal to $\Sigma(\Omega)$, and the input and output alphabets of the corresponding X/Y-automaton are of the form $X = \Sigma(U)$ and $Y = \Sigma(W)$. Thus, any vector $\sigma \in \Sigma(\Omega)$ can be considered as a pair $< x, y >$, where $x \in X$ and $y \in Y$. Therefore, along with the designation $\langle \sigma_0, \sigma_1, \dots, \sigma_r \rangle$ for a state of depth r, we will use the designation $\langle < x_0, y_0 >, \dots, < x_r, y_r > \rangle$. For this representation of a state $q \in Q(r,\Omega)$, we define one more region of the space of states that is connected with the state q. In a set $N(q) = \{\langle < x_1, y_1 >, \dots, < x_r, y_r >, < x, y > \rangle \mid x \in X,\ y \in Y\}$, we single out a subset of states by assuming that x is equal to x'. We call this subset the region of transition from the state q under the action of the symbol x' and denote this subset by $N(q, x')$.

There are several methods for defining automata semantics of a specification in the language L. We will consider two of them that are used in this article to establish the correspondence between properties of specifications and the automata specified by them. These methods are based on the concept of the state space for a formula $f(t)$ in the specification $F=\forall tf(t)$ and differ in the length of the word used in the capacity of a state of the automaton being specified. In the first method, states of an automaton are identified with states from $Q(r,\Omega)$, i.e., with words of length $r+1$ in the alphabet $\Sigma(\Omega)$. The Σ-automaton $A^{r+1}(f(t))=\langle\Sigma(\Omega),Q^{r+1},\delta^{r+1}\rangle$ is first defined, where the set of states Q^{r+1} consists of all the states from $Q(r,\Omega)$ on which $f(t)$ is true. The transition function δ^{r+1} is defined as follows. Assume that a state $q\in Q^{r+1}$ is of the form $\langle\sigma_0,\sigma_1,\dots,\sigma_r\rangle$ and that $\sigma\in\Sigma(\Omega)$; then the value of $\delta^{r+1}(q,\sigma)=\langle\sigma_1,\dots,\sigma_r,\sigma\rangle$ if $\langle\sigma_1,\dots,\sigma_r,\sigma\rangle$ belongs to Q^{r+1} and is not defined otherwise. The automaton $A^{r+1}(F)$ defining the semantics of the specification $F=\forall tf(t)$ is the maximal cyclic subautomaton of the automaton $A^{r+1}(f(t))$ and is also called the kernel of this automaton.

In the second method for defining the semantics, states of an automaton are identified with words of length r in the alphabet $\Sigma(\Omega)$. A Σ-automaton $A^r(f(t))=\langle\Sigma(\Omega),Q^r,\delta^r\rangle$ is first considered, where states from Q^r correspond to transition regions for the states of the space $Q(r,\Omega)$ on which the formula $f(t)$ is true. Let the transition region be specified by the formula $S(t-1)$ (minterm of all atoms whose rank is less than zero); then the state $q=\langle\sigma_1,\dots,\sigma_r\rangle$ corresponding to it determines a collection of values of atoms $p_1(t-1),\dots,p_m(t-1),\dots,p_1(t-r),\dots,p_m(t-r)$ on which this formula is true.

Let a state $q=\langle\sigma_1,\dots,\sigma_r\rangle$ belong to Q^r, and let $\sigma\in\Sigma(\Omega)$. Then the value of $\delta^r(q,\sigma)$ is defined and is equal to $\langle\sigma_2,\dots,\sigma_r,\sigma\rangle$ if $\langle\sigma_1,\dots,\sigma_r,\sigma\rangle\in Q^{r+1}$; otherwise, the value of $\delta^r(q,\sigma)$ is not defined. The automaton $A^r(F)$ defining the semantics of the specification $F=\forall tf(t)$ is the maximal cyclic subautomaton (kernel) of the automaton $A^r(f(t))$.

Let us show the equivalence of the described methods of definition of semantics. To this end, we will define a mapping $\gamma: Q^{r+1}\to Q^r$ as follows: $\gamma(\langle\sigma_0,\sigma_1,\dots,\sigma_r\rangle)=\langle\sigma_1,\dots,\sigma_r\rangle$.

It follows from the definitions of the mapping γ and transition function δ^r that if, for $\langle\sigma_0,\sigma_1,\dots,\sigma_r\rangle\in Q^{r+1}$, the value of $\delta^{r+1}(\langle\sigma_0,\sigma_1,\dots,\sigma_r\rangle,\sigma)$ is defined, then $\delta^r(\gamma(\langle\sigma_0,\sigma_1,\dots,\sigma_r\rangle),\sigma)$ is also defined and is equal to $\langle\sigma_2,\dots,\sigma_r,\sigma\rangle$, i.e., $\gamma(\delta^{r+1}(\langle\sigma_0,\sigma_1,\dots,\sigma_r\rangle,\sigma))=\delta^r(\gamma(\langle\sigma_0,\sigma_1,\dots,\sigma_r\rangle),\sigma)$. Thus, γ is a homomorphism of the set of states of the automaton $A^{r+1}(f(t))$ into the set of states of the automaton $A^r(f(t))$ that preserves the transition function of the automata. Note that the states from Q^r that do not belong to $\gamma(Q^{r+1})$, i.e., do not have preimages, cannot be states of the cyclic automaton $A^r(F)$. This implies that the automata $A^{r+1}(F)$ and $A^r(F)$ are equivalent (strictly equivalent).

REPRESENTATION OF SPECIFICATIONS AND A REZOLUTION PROCEDURE

The methods of harmonization of automata specifications that are considered below are based on the representation of specifications in the form of sets of clauses. In the language L, a clause is understood to be the disjunction of literals of the form $\tilde{p}(t-k)$, where $\tilde{p}\in\{p,\neg p\}$, $k\in\mathbf{N}$. A clause that does not contain literals is called empty. A set of clauses C is considered as the specification of the formula $\forall tf(t)$, where $f(t)$ is the conjunction of all clauses from C. A clause is called normalized to the right if the maximum rank of its atoms is equal to zero. We call the clause obtained from a clause $c(t)$ by increasing the rank of all its atoms by $k\in\mathbf{N}$ the k-shift of $c(t)$ to the right and denote it by $c(t+k)$. A decrease in the rank of all atoms in a clause is called its shift to the left. The replacement of an unnormalized clause $c(t)$, i.e., a clause in which the maximum rank of atoms is less than zero, by its shift to the right with the maximum rank of atoms equal to zero is called the normalization of the clause. In what follows, sets of clauses normalized to the right are considered as specifications of automata.

In specifying a formula $f(t)$ in the specification $F=\forall tf(t)$ by a set of clauses C, we denote by $A(r,C)$ the automaton $A(f(t))$ defined by the formula $f(t)$ in the state space $Q(r,\Omega)$, where r is the depth of the formula $f(t)$. To refine the type of semantics, the superscripts $r+1$ or r will be sometimes added in designations of automata.

Let c_1 and c_2 be clauses normalized to the right, let $p(t)$ be an atom of zero rank, let $c_1=c_1'\vee p(t)$, and let $c_2=c_2'\vee\neg p(t)$. The clause $c=c_1'\vee c_2'$ is called the R-resolvent of clauses c_1 and c_2 with respect to the atom $p(t)$.

The operation of obtaining an R-resolvent of two clauses is called resolution. By the R-deduction of a clause c from a set of clauses C we understand a finite sequence of clauses $c_1, \ldots, c_k$ such that $c_k = c$ and each clause c_i $(i = 1, \ldots, k)$ belongs to C, or is an R-resolvent of two previous clauses, or is the result of normalization of the clause c_{i-1} to the right.

The following statement is true. The formula $\forall t f(t)$ specified by a set C of clauses normalized to the right is inconsistent if and only if there is an R-deduction of the empty clause from C.

A clause $c_1(t)$ subsumes the clause $c_2(t)$ if there is some $k \in \mathbf{Z}$ such that all literals of the clause $c_1(t+k)$ are present in the clause $c_2(t)$. The elimination of the clauses subsumed by other clauses of the set of clauses that represents the formula $\forall t f(t)$ from this set leads to a set of clauses specifying an equivalent formula (i.e., a formula that has the same set of models).

A set C of clauses normalized to the right is called R-complete if, for any R-resolvent of clauses c_1 and c_2 from C, it contains a clause subsuming this resolvent. In other words, the set C is closed with respect to resolving. A procedure that, for a given set of clauses C, constructs an R-complete set equivalent to it is called the R-completion of the set of clauses C. It is one of basic procedures used to harmonize automata specifications.

Let us consider a property of an R-complete set of clauses that will be used in what follows. Let $f(t)$ be the formula defined by a set of clauses C. We denote by $P(f(t))$ the formula specifying the set of all states from $Q(r, \Omega)$ that immediately precede the states specified by the formula $f(t)$. The elimination of all dead states in the automaton $A^{r+1}(f(t))$ corresponds to the construction of the automaton $A^{r+1}(f(t) \& P(f(t)))$. Note that new dead states can arise in the automaton $A^{r+1}(f(t) \& P(f(t)))$. As is shown in [5], the set of clauses specifying the formula $f(t) \& P(f(t))$ is obtained by closing the set of clauses C with respect to the resolution (with respect to atoms of zero rank) and adding the clauses obtained as a result of normalization of all existing unnormalized clauses. Iterative execution of this procedure leads to the set of clauses specifying the formula $f^*(t)$ such that the formula $\forall t f^*(t)$ is equivalent to the formula $\forall t f(t)$ and the automaton $A^{r+1}(f^*(t))$ does not contain dead states.

R-completion is characterized by the fact that, at each iteration, the clauses that do not contain zero-rank literals are eliminated (subsumed). In this case, on some states of the region specified by the negation of an eliminated clause (on which the initial formula $f(t)$ is false), the obtained formula can become true. Let $c(t)$ be a clause without zero-rank literals, and let $c(t+k)$ be the corresponding normalized clause. In the state space, the negation of this clause specifies the set of states from which the region specified by $\neg c(t)$ is reachable in k steps. The formula obtained as a result of subsumption of the clause $c(t)$ is false on the states from $\neg c(t+k)$ and, hence, the states added as a result of this subsumption to the automaton $A(r, C)$ are not reached from the kernel of this automaton. Thus, the following property of an R-completed set of clauses is true.

Statement 1. The set of states specified by an R-complete set of clauses in the corresponding state space consists of its kernel (cyclic automaton) and, possibly, states that are not reachable from the kernel.

HARMONIZATION OF SPECIFICATIONS

In considering automata harmonization, the most convenient type of composition of automata is their parallel composition to which corresponds the conjunction of specifications, i.e., the union of sets of clauses. Let us recall the main idea of automata harmonization using parallel composition. In this case, in such a composition, instead of an automaton B with which an automaton A is harmonized, its left shift with respect to its input $\widetilde{B}$ is used [2]. The specification of the automaton $\widetilde{B}$ is obtained from the specification of the automaton B by decreasing the ranks of all output atoms, i.e., the atoms formed by the output predicate symbols (for the automaton A), by one. The specification of the automaton B is often initially given in this form.

For the Σ-automaton corresponding to the parallel cyclic composition of an X/Y-automaton A and a Y/X-automaton B that are considered as Σ-automata, the consistency condition is fulfilled if, at each of its state (q, s), the projection of the set of conditions of all transitions from this state onto X coincides with the projection of the set of conditions of all transitions from a state s onto X in the Σ-automaton B. As applied to the state (q, s), this condition is called the correctness condition of the composition at the state (q, s) [2].

The process of harmonization of an automaton A with an automaton B, which is described in [2], is executed as follows. The parallel cyclic composition of the Σ-automata A and $\widetilde{B}$ is constructed. Then the states of this composition are checked for the correctness condition. All the states of this composition that do not satisfy the correctness condition (incorrect states) are eliminated. In the obtained automaton, the maximal cyclic subautomaton is again singled out, and its states are checked for the correctness condition. This process continues until a cyclic subautomaton is obtained in which all states satisfy the correctness condition or all the states of the composition are eliminated. The latter testifies to the impossibility of harmonization of the automaton A with the automaton B.

Let us consider how this process can be realized in the case of harmonization of specifications of the Σ-automata A and $\widetilde{B}$ represented by sets of clauses C_A and $C_{\widetilde{B}}$ with the same set of predicate symbols Ω. Assuming that the set Ω is ordered by the same method for both specifications, we will consider the automata $A(r_A, C_A)$ and $A(r_{\widetilde{B}}, C_{\widetilde{B}})$ and the parallel cyclic composition of the automata A and $\widetilde{B}$ in the same state space $Q(r, \Omega)$, where $r = \max(r_A, r_{\widetilde{B}})$. It is easy to show that if a state (q, s) belongs to the parallel cyclic composition of the automata A and $\widetilde{B}$ (i.e., to the cyclic automaton), then to the states q, s, and (q, s) corresponds the same state of the state space being considered. The truth or falsity of the correctness condition for the state (q, s) is determined by the properties of the transition region $N((q, s))$ for this states. Since the state (q, s) belongs to the cyclic automaton, the functions $f_A(t)$ and $f_{\widetilde{B}}(t)$ specified by the sets of clauses C_A and $C_{\widetilde{B}}$, respectively, are true on the transition region $N((q, s))$. The state (q, s) is incorrect if and only if $N((q, s))$ contains the transition region that corresponds to some $x \in X$ and, on this region, the function specified by the set of clauses $C_A \cup C_{\widetilde{B}}$ is false and $f_{\widetilde{B}}(t)$ is true. This implies that the state (q, s) is incorrect only when the state of the automaton $A(r, C_A \cup C_{\widetilde{B}})$ considered as an X / Y-automaton is partial, i.e., the transition from (q, s) under the action of x is not defined. Therefore, the process of harmonization of specifications can be carried out as follows. The transformation of the set of clauses $C_A \cup C_{\widetilde{B}}$ is performed that corresponds to singling out the maximal cyclic automaton that is a subautomaton of the automaton $A(r, C_A \cup C_{\widetilde{B}})$ and coincides with the parallel cyclic composition of the automata A and $\widetilde{B}$. Then partial states of this automaton considered as an X / Y-automaton are determined, i.e., the transition regions are found on which the formula specified by the set of clauses $C_A \cup C_{\widetilde{B}}$ is false. On these transition regions, the truth value of the formula $f_{\widetilde{B}}(t)$ is determined. If incorrect states are revealed, then the transformation of the specification $C_A \cup C_{\widetilde{B}}$ is performed that corresponds to the elimination of this states. In the automaton specified by the obtained set of clauses, the maximal cyclic subautomaton is again singled out, etc. Note that, in the described process of harmonization, it suffices to restrict ourselves to singling out the maximal quasicyclic subautomaton, which is performed by the procedure of completion of the corresponding set of clauses. As will be shown below, the same procedure is used for finding partial states. In this case, the partial states that do not belong to the kernel of the automaton $A(r, C_A \cup C_{\widetilde{B}})$ can be analyzed, which does not affect the result of execution of the algorithm.

PARTIALNESS

We will formulate the condition of elimination of states in terms of properties of sets of clauses. To this end, we introduce the following concepts. In partitioning a set of predicate symbols Ω into input and output symbols, we call the clauses normalized to the right and containing output literals of zero rank first-kind clauses and the clauses that do not contain output literals of zero rank second-kind clauses.

Let $\Omega = \{p_1, \ldots, p_m\}$. Denote by $S_\Omega(t-k)$ the set of all elementary conjunctions of the form $\widetilde{p}_1(t-k) \& \ldots \& \widetilde{p}_m(t-k)$, where $k \in \mathbf{N}$, $\widetilde{p}_i \in \{p_i, \neg p_i\}$, and $p_i \in \Omega$ $(i = 1, \ldots, m)$. A state $q = \langle \sigma_0, \sigma_1, \ldots, \sigma_r \rangle$ from $Q(r, \Omega)$ is specified by the formula $f_q(t) = s_0(t-r) \& \ldots \& s_r(t)$, where $s_i(t-r+i) \in S_\Omega(t-r+i)$ and assumes the value 1 on a vector σ_i $(i = 0, \ldots, r)$. The transition region in the state space $Q(r, \Omega)$ is specified by the conjunction of the form $s_0(t-r) \& \ldots \& s_{r-1}(t-1)$, where $s_i(t-r+i) \in S_\Omega(t-r+i)$.

Let $d(t-1)$ be the conjunction specifying a transition region, and let $d_0(t)$ be the conjunction of literals that is not identically equal to zero and is formed by atoms $p_{i_1}(t), \ldots, p_{i_k}(t)$, $k \le m$, belonging to a set $\Omega_0(t) = \{p_1(t), \ldots, p_m(t)\}$. Let $f(t)$ be a formula of signature Ω, and let it be specified by a set of clauses $C = \{c_1(t), c_2(t), \ldots, c_n(t)\}$.

Statement 2. If $f(t)$ is false on the domain specified by the formula $d(t-1)\& d_0(t)$, i.e., $d(t-1)\& d_0(t)\& f(t)\equiv 0$, then the closure of C with respect to resolving upon atoms of zero rank from $\Omega_0(t)\setminus\{p_{i_1}(t),\dots,p_{i_k}(t)\}$ contains a clause $c(t)$ that is false on this domain.

Proof. Let us represent the conjunction $d(t-1)\& d_0(t)\& f(t)=d(t-1)\& d_0(t)\& \bigwedge_{i=1}^{n} c_i(t)$ in the form $d(t-1)\& d_0(t)\& c_1^*(t)\&\dots\& c_g^*(t)$, where each clause $c_j^*(t)$, $j=1,\dots,g$, is a part of some $c_i(t)\in C$ consisting of all its literals formed by atoms of a set $\Omega_0(t)\setminus\{p_{i_1}(t),\dots,p_{i_k}(t)\}$. A set of clauses $C^*=\{c_1^*(t),\dots,c_g^*(t)\}$ is constructed as follows. All clauses containing literals available in $d(t-1)\& d_0(t)$ are first eliminated from C. Any clause $c_i(t)$ can be eliminated from C since $d(t-1)\& d_0(t)\& c_i(t)\Leftrightarrow d(t-1)\& d_0(t)$. Then all literals whose negations are contained in $d(t-1)\& d_0(t)$ are eliminated in the remained clauses. As a result, the set C^* is obtained whose clauses contain only literals from $\Omega_0(t)\setminus\{p_{i_1}(t),\dots,p_{i_k}(t)\}$.

The formula $d(t-1)\& d_0(t)\& c_1^*(t)\&\dots\& c_g^*(t)$ is inconsistent (is identically equal to zero) if and only if $c_1^*(t)\&\dots\& c_g^*(t)$ is inconsistent, which, in turn, is possible if and only if the empty clause is deduced from C^* [4]. Since to each clause from C^* corresponds a clause whose part the former close is in C, to the deduction of the empty clause from C^* corresponds a deduction of a clause from C that does not contain atoms from $\Omega_0(t)\setminus\{p_{i_1}(t),\dots,p_{i_k}(t)\}$ and whose all literals are negations of literals appearing in the conjunction $d(t-1)\& d_0(t)$. This clause is false on the domain $d(t-1)\& d_0(t)$ and belongs to the closure of C with respect to resolving upon atoms of zero rank from $\Omega_0(t)\setminus\{p_{i_1}(t),\dots,p_{i_k}(t)\}$.

COROLLARY. If C is closed with respect to resolving upon output atoms of zero rank, then a necessary condition of the partialness of the automaton $A(r,C)$ is the presence of second-kind clauses in C.

In fact, a state q is partial if and only if there is a transition region for it on which the formula $f(t)$ is false. Let $d_0(t)$ be an elementary conjunction specifying such a transition region. As is obvious, $d_0(t)$ does not contain output literals of zero rank and, hence, a clause $c(t)\in C$ such that $d_0(t)\& c(t)\equiv 0$ also does not contain them, i.e., it is a second-kind clause. This condition is not sufficient since the transition region on which the formula $f(t)$ is false is not necessarily the transition region for a state of the automaton $A(r,C)$.

Note that the condition of partialness of states of the automaton $A^r(f(t))$ is the same as for the automaton $A^{r+1}(f(t))$ (the presence of a transition region on which $f(t)$ is false). However, whereas, for the automaton $A^{r+1}(f(t))$, partialness pertains to the states from which there is a transition to the corresponding transition region, for the automaton $A^r(f(t))$, partialness pertains to a state corresponding to this transition region.

HARMONIZATION PROCEDURES

To harmonize an automaton A with an automaton B, it is not necessary to single out the maximal cyclic subautomaton in the automaton $A(r, C_A\cup C_{\tilde{B}})$, but it is only necessary to successively eliminate incorrect states from it. However, to this end, it is necessary that, for all the states reachable from the kernel of this automaton, the transition function should be defined. As Statement 1 implies, this can be achieved by completing the set of clauses $C_A\cup C_{\tilde{B}}$. The procedure of harmonization is as follows. The set $C_A\cup C_{\tilde{B}}$ is R-completed. The set of clauses $C'_{\tilde{B}}$ is simultaneously formed by the addition of all the clauses that do not contain literals of zero rank that are deduced from $C_A\cup C_{\tilde{B}}$ to $C_{\tilde{B}}$. Let us give some explanations concerning the set $C'_{\tilde{B}}$. In the course of harmonization, the behaviour of the automaton $A(r, C_{\tilde{B}})$ is important only at the states corresponding to the kernel of the automaton $A(r, C_A\cup C_{\tilde{B}})$. Recall that they are the same states in the state space being considered. From such states, transitions are possible only to the transition regions corresponding to r-states of the kernel of the automaton $A^r(r, C_A\cup C_{\tilde{B}})$. Thus, any clause whose negation specifies a domain nonintersecting with the transition regions for $(r+1)$-states of the kernel of the automaton

$A(r, C_A \cup C_{\widetilde{B}})$ can be added to $C_{\widetilde{B}}$, though, in this case, the automaton $A(F_{\widetilde{B}})$ being specified can change. This addition can reduce the number of clauses generated in analyzing the transition regions corresponding to second-kind clauses that are deduced from $C_A \cup C_{\widetilde{B}}$. Proceeding from the same considerations, such left shifts of any clause that is deducible from $C_A \cup C_{\widetilde{B}}$ and in which the minimal rank of atoms is no less than $-r$ can be added to $C_{\widetilde{B}}$.

The negation of each second-kind clause c obtained in completing $C_A \cup C_{\widetilde{B}}$ specifies the set of transition regions on which the formula defined by the set of clauses $C_A \cup C_{\widetilde{B}}$ is false. To reveal the clauses that correspond to incorrect states, $\neg c$ is multiplied by the conjunction of clauses from $C'_{\widetilde{B}}$. Here, it is pertinent to use the same expedient as in constructing the set C^* in the proof of Statement 2. The clauses containing a literal appearing in $\neg c$ are first eliminated from $C'_{\widetilde{B}}$, the literals that are negations of literals from $\neg c$ are eliminated from the remained clauses, and then $\neg c$ and the remained clauses are multiplied. If all literals are eliminated from some clause, then the product being computed is identically equal to zero. The obtained product specifies the set of all the states from the transition regions being considered on which the formula $f_{\widetilde{B}}(t)$ is true. The transition regions for incorrect states are determined in this manner. To obtain the formula specifying the states for which these regions are transition regions, it is necessary to eliminate all zero-rank literals from an arbitrary DNF of the obtained product and then to increase the ranks of all the remained literals by one. We denote this formula by $f_c(t)$. To the elimination of the states specified by the formula $f_c(t)$ from the automaton $A(r, C_A \cup C_{\widetilde{B}})$ corresponds the addition of the set of clauses defined by an arbitrary CNF of the formula $\neg f_c(t)$ to $C_A \cup C_{\widetilde{B}}$. If some of the added clauses are not normalized to the right, then they should be normalized.

After the execution of the described procedure of elimination of incorrect states for all second-kind clauses and, as a result, new clauses will be possibly added to the set $C_A \cup C_{\widetilde{B}}$, the obtained set of clauses is R-completed. If new second-kind clauses appear as a result of completion, then the process is repeated until the empty clause is obtained after the next R-completion, which testifies to the inconsistency of the automaton A with the automaton B, or all the newly obtained second-kind clauses will be subsumed by already available clauses. At this point, the process of elimination of incorrect states and, hence, harmonization, comes to an end. The result of harmonization is the X/Y-automaton specified by the obtained set of clauses.

Example 1. The automaton A is specified by the following set of clauses C_A:

#1 $(\neg u(t-1) \vee \neg w(t))$,
#2 $(u(t) \vee \neg w(t))$,
3 $(u(t-1) \vee u(t))$,
#4 $(\neg u(t-1) \vee \neg w(t-1) \vee \neg u(t))$,

where u is the input predicate symbol and w is the output one. The set of clauses $C_{\widetilde{B}}$ specifying the automaton $\widetilde{B}$ is as follows:

5 $(\neg u(t-1) \vee w(t-1) \vee u(t))$,
6 $(w(t-2) \vee w(t-1) \vee u(t))$,
7 $(w(t-2) \vee \neg u(t-1) \vee u(t))$,
#8 $(\neg w(t-2) \vee u(t-1) \vee w(t-1) \vee \neg u(t))$,
#9 $(w(t-2) \vee u(t-1) \vee \neg w(t-1) \vee \neg u(t))$.

The process of completion of the set $C_A \cup C_{\widetilde{B}}$ is as follows.

Clause 9 is subsumed by clause 2. The subsumed clauses are labeled by the symbol #. Clause 2 is resolved upon the atom $u(t)$ with clauses 4 and 8 and yields, respectively, the clauses

#10 $(\neg u(t-1) \vee \neg w(t-1) \vee \neg w(t))$ is subsumed by clause 1,
#11 $(\neg w(t-2) \vee u(t-1) \vee w(t-1) \vee \neg w(t))$.

Clause 3 is resolved with clause 8 and yields unnormalized clause 12 that is normalized, which is denoted by the symbol =>,

#12 $(\neg w(t-2) \vee u(t-1) \vee w(t-1)) \Rightarrow (\neg w(t-1) \vee u(t) \vee w(t))$.

Clause 12 subsumes clauses 8 and 11. Clause 4 is resolved with clause 7 and yields clause 13 that is normalized,

#13 $(w(t-2) \vee \neg u(t-1) \vee \neg w(t-1)) \Rightarrow (w(t-1) \vee \neg u(t) \vee \neg w(t))$.

Resolving clause 1 with clause 12 yields

#14 $(\neg u(t-1) \vee \neg w(t-1) \vee u(t))$.

Resolving clauses 2 and 12 yields clause #15 $(\neg w(t-1) \vee u(t))$ subsuming clauses 12 and 14.

Clauses 2 and 13 yield clause #16 $(w(t-1) \vee \neg w(t))$ subsuming clause 13.

Clauses 4 and 15 yield clause #17 $(\neg u(t-1) \vee \neg w(t-1)) => (\neg u(t) \vee \neg w(t))$.

Clauses 2 and 17 yield clause 18 $(\neg w(t))$ that subsumes clauses 1, 2, 4, 10, 15, 16, and 17. At this point, the process of completion terminates. As a result, we have the following completed set of clauses:

3 $(u(t-1) \vee u(t))$,

5 $(\neg u(t-1) \vee w(t-1) \vee u(t))$,

6 $(w(t-2) \vee w(t-1) \vee u(t))$,

7 $(w(t-2) \vee \neg u(t-1) \vee u(t))$,

18 $(\neg w(t))$.

This set has only one second-kind clause $(u(t-1) \vee u(t))$ that does not belong to $C_{\widetilde{B}}$.

In the course of completion, the following set $C'_{\widetilde{B}}$ is formed:

$(\neg u(t-1) \vee w(t-1) \vee u(t))$,

$(w(t-2) \vee w(t-1) \vee u(t))$,

$(w(t-2) \vee \neg u(t-1) \vee u(t))$,

$(\neg w(t-1))$, $(\neg w(t-2))$.

The result of multiplying the formula $\neg u(t-1) \neg u(t)$ specified by the negation of clause 3 by the conjunction of clauses from the set $C'_{\widetilde{B}}$ is identically equal to zero. Therefore, the process of harmonization terminates, and the completed set of clauses $C_A \cup C_{\widetilde{B}}$ can be considered as the specification of the automaton harmonized with the automaton B.

We now consider an implementation of this method of harmonization of an X/Y-automaton A with a Y/X-automaton B without analyzing the partialness of states of the X/Y-automaton $A(r, C_A \cup C_{\widetilde{B}})$. In this case, the automata $A^r(r, C_A \cup C_{\widetilde{B}})$ and $A^r(r, C_{\widetilde{B}})$ are considered. As above, the kernel of the automaton $A(r, C_A \cup C_{\widetilde{B}})$ does not singled out, and, to eliminate dead $(r+1)$-states reachable from it, the set of clauses $C_A \cup C_{\widetilde{B}}$ is completed. Moreover, in order to considerably decrease the number of states being analyzed in the automaton $A^r(r, C_A \cup C_{\widetilde{B}})$, the completed set of clauses is extended by clauses available and shifted to the left so that the rank of each atom in a shift is no less than the minimal rank of atoms formed by the same predicate symbol in $C_A \cup C_{\widetilde{B}}$. This does not change the kernel of the automaton being specified. With the same purpose, the set of clauses $C'_{\widetilde{B}}$ is formed as is described above. The conjunction of all clauses of the extended set $C_A \cup C_{\widetilde{B}}$ is transformed into a DNF. In each elementary conjunction of this DNF, its part formed by literals whose rank is less than zero specifies the set of states of the automaton $A^r(r, C_A \cup C_{\widetilde{B}})$, and its part formed by zero-rank literals specifies the same set of conditions of transitions for all these states. Let l be one of such conjunctions. Eliminating zero-rank literals from it and multiplying the remained part by the conjunction of all clauses from $C'_{\widetilde{B}}$, we obtain one or several conjunctions whose parts formed by literals of zero rank specify sets of conditions of transitions from the corresponding states of the automaton $A^r(r, C_{\widetilde{B}})$. The projection of the set of conditions of transitions onto X is obtained if all the literals formed by output predicate symbols are eliminated from the conjunction specifying it. Comparing projections of the obtained sets of conditions of transitions onto X with the projection of sets of conditions of the transitions specified by the conjunction l onto X, we determine the presence of incorrect states among the states specified by the conjunction l. They are eliminated by multiplying the DNF of the formula corresponding to the set of clauses $C_A \cup C_{\widetilde{B}}$ extended by the negations of the conjuncts specifying incorrect states. The described process is iteratively executed until incorrect states cease to appear or all the states of the automaton being considered are eliminated.

Example 2. Let us consider the harmonization of the automata that are the same as in example 1.

The completed set of clauses $C_A \cup C_{\widetilde{B}}$ is as follows:

3 $(u(t-1) \vee u(t))$,

5 $(\neg u(t-1) \vee w(t-1) \vee u(t))$,
6 $(w(t-2) \vee w(t-1) \vee u(t))$,
7 $(w(t-2) \vee \neg u(t-1) \vee u(t))$,
18 $(\neg w(t))$.

We augment it with the following shifts of clause 18 to the left: $(\neg w(t-1))$ and $(\neg w(t-2))$. The product of all these clauses represented in DNF is $\neg w(t-2) \neg w(t-1) u(t) \neg w(t)$. It specifies two equivalent r-states such that the set of conditions of transitions from these states is defined by the formula $u(t) \neg w(t)$. The set $C'_{\widetilde{B}}$ together with the added shifts of clauses is as follows:

$(\neg u(t-1) \vee w(t-1) \vee u(t))$,
$(w(t-2) \vee w(t-1) \vee u(t))$,
$(w(t-2) \vee \neg u(t-1) \vee u(t))$,
$(\neg w(t-1))$, $(\neg w(t-2))$.

The multiplication of the formula $\neg w(t-2) \neg w(t-1)$ specifying the two obtained r-states by the conjunction of clauses from $C'_{\widetilde{B}}$ yields $\neg w(t-2) \neg w(t-1) u(t)$. The set of conditions of transitions from the states being considered in the automaton $A^r(r, C'_{\widetilde{B}})$ is defined by the formula $u(t)$. It is easy to see that the projections of sets of conditions of transitions from the states being considered in the automata $A^r(r, C_A \cup C_{\widetilde{B}})$ and $A^r(r, C_{\widetilde{B}})$ onto the input alphabet $X = \{u, \neg u\}$ coincide and are equal to $u(t)$. Thus, these states are correct. This implies that the automaton specified by the extended set of clauses $C_A \cup C_{\widetilde{B}}$ is harmonized with the automaton B. This cyclic automaton has one state that is specified by the formula $\neg w(t-2) \neg w(t-1) u(t-1)$ and that, under the action of the input symbol u, transits to itself and produces the symbol $\neg w$.

Note that the cardinality of the state space $Q(r, \Omega)$ considered in practice is not always equal to $2^{|\Omega|(r+1)}$ and is determined by minimal ranks of atoms generated by each of predicate symbols. In particular, in the considered example, $r = 2$ and $|\Omega| = 2$, and the cardinality of the state space is equal not to $2^{2\times 3} = 64$ but to $2^3 \times 2^2 = 32$. This is conditioned by the absence of the atom $u(t-2)$ in the specification.

CONCLUSIONS

There can be two approaches to organizing the process of harmonization of specifications of interacting automata A and B represented by sets of clauses C_A and C_B ($C_{\widetilde{B}}$), respectively. Both approaches are based on the method of harmonization of automata that uses their parallel cyclic composition.

In the first approach described in [3], the specification C_A of the automaton being harmonized is transformed with the added first-kind clauses deduced from $C_{\widetilde{B}}$ after the R-completion of it. They are additional statements imposing constraints on the behaviour of the automaton A. $C_{\widetilde{B}}$ is completed only once, and then the obtained set does not change.

In the second approach considered in the present article, the specification of the parallel composition of automata A and $\widetilde{B}$, i.e., $C_A \cup C_{\widetilde{B}}$ is transformed. The process of harmonization begins with the R-completion of the set $C_A \cup C_{\widetilde{B}}$. In this case, a much larger set of clauses than in the first case should be completed, but the subsequent transformations are executed over a simpler automaton and the number of tests of states for correctness decreases.

In both approaches, a specification does not undergo the transformation corresponding to singling out the maximal cyclic subautomaton in the automaton $A(r, C)$, where C is the set of clauses being considered, though the process of harmonization consists of eliminating incorrect states exactly from this automaton. As has been shown, the R-completion of the set of clauses C intrinsically corresponds to singling out the quasicyclic automaton from the automaton $A(r, C)$, which increases the number of states being analyzed since not only the correctness of states of the cyclic subautomaton is checked but also the correctness of the obtained result is preserved.

This paper considers two methods for detecting and eliminating incorrect states at the level of specifications. In the first method, the analysis of correctness is restricted to partial states as is described in [3]. It should be noted that both detecting partial states and singling out the quasicyclic subautomaton in the automaton $A(r, C)$ are carried out by a procedure of completion of the set of clauses C. Completion is performed at each iteration of the execution of the algorithm after the addition of clauses that corresponds to the elimination of incorrect states. At each of such stages except for the initial one, an already completed set with several clauses added to it is completed. This simplifies the process of completion. Statement 2 makes it possible to use not the completion of a set of clauses for finding partial states but its closure with respect to resolving upon output atoms of zero rank. However, since a quasicyclic subautomaton should be singled out in the automaton $A(r, C)$, R-completion is performed in which resolving is performed upon all atoms of zero rank. Moreover, in completing the set of clauses $C_A \cup C_{\tilde{B}}$, it is often convenient to use the separable resolution method [6] in which the partition of predicate symbols into input and output symbols is used to decrease the number of generated clauses.

In the second method, all the states of the automaton $A(r, C)$ that are represented by the disjunctive normal form of a formula $f(t)$ are analyzed. Here, to decrease the number of states being analyzed, all possible shifts of clauses of the set C to the left that preserve the transformed formula $f(t)$ within the framework of the initial state space are added to this set. The addition of such clauses eliminates the states that do not belong to the cyclic subautomaton of the automaton $A(r, C)$.

This article describes two methods of definition of the semantics of the language L that differ in the length of words representing states of the automaton being specified in the alphabet $\Sigma(\Omega)$. The use of words of length r decreases the space in which states of the automaton $A^r(r, C)$ are located by a factor of $2^{|\Omega|}$ and, hence, the number of the states being analyzed decreases accordingly. Therefore, it is convenient to use this semantics of the language L in the second of the described methods.

The expediency of applying each of the mentioned three methods of harmonization of specifications is determined by the type of the initial specification of interacting automata. Note that is also expedient to add shifts of clauses to the left to $C_{\tilde{B}}$ in the method described in [3].

REFERENCES

1. A. N. Chebotarev, "Regular form of deterministic FSMs specifications in the language L," Prikl. Diskr. Matemetika., No. 4, 64–72 (2010).
2. A. N. Chebotarev, "Harmonization of interacting automata," Cybernetics and Systems Analysis, **51**, No. 5, 676–686 (2015).
3. A. N. Chebotarev, "General method of testing the compatibility of interacting automata with finite memory," Cybernetics and Systems Analysis, **35**, No. 6, 867–874 (1999).
4. C.-L. Chang and R. C.-T. Lee, Symbolic Logic and Mechanical Theorem Proving [Russian translation], Nauka, Moscow (1983).
5. A. N. Chebotarev, "Consistency test for simple specifications of automaton systems," Cybernetics and Systems Analysis, **30**, No. 3, 317–322 (1994).
6. A. N. Chebotarev, "Separable resolution method for checking the satisfiability of formulas in the language L," Cybernetics and Systems Analysis, **34**, No. 6, 794–799 (1998).