Temporal Verification of Probabilistic Multi-Agent Systems ⋆

Michael I. Dekhtyar[1], Alexander Ja. Dikovsky[2], and Mars K. Valiev[3]

[1] Dept. of CS, Tver St. Univ. Tver, Russia, 170000, Michael.Dekhtyar@tversu.ru
[2] LINA, Université de Nantes, France, Alexandre.Dikovsky@irin.univ-nantes.fr
[3] Keldysh Inst. for Appl. Math. Moscow, Russia, 125047, valiev@spp.keldysh.ru

To Boris Avraamovich, our teacher in life and research, on occasion of his 85th Anniversary

Abstract. Probabilistic systems of interacting intelligent agents are considered. They have two sources of uncertainty: uncertainty of communication channels and uncertainty of actions. We show how such systems can be polynomially transformed to finite state Markov chains. This allows to transfer known results on verifying temporal properties of the finite state Markov chains to the probabilistic multi-agent systems of considered type.

1 Introduction

Last time there has been increasing interest in the area of software multi-agent systems (MAS). The range of applications of MAS is very broad and extends from operating system interfaces, processing of satellite imaging data and WEB navigation to air traffic control, business process management and electronic commerce. The states and interaction rules of agents in MAS may be very complicate. This makes the behavior of MAS (as well as of other concurrent software systems) badly predictable and leads to necessity of developing formal means to analyze this behavior.

There is a number of papers on this matter in the literature which deal with different models of agents, multi-agent systems and specification languages describing their behavior. In particular, in [17, 19] a behavior is considered for abstract agents with no internal structure, in [3, 11] agents are specified by formulas of some temporal logics. Another popular approach to describing agents is based on "Believe-Desire-Intention" model initiated in [14] (see also [4, 5, 20]. In our previous papers [8, 9] we considered verification complexity for MAS constructed on the base of IMPACT-architecture introduced in [16].

In all these papers it is assumed that all agents operate with a complete and certain view of the world, and information transfer from one agent to another

⋆ This work was sponsored by the Russian Fundamental Studies Foundation (Grants 07-01-00637-a and 08-01-00241-a).

is lossless and takes some determined time. However, in many real-world applications, these assumptions are not satisfied, and agents have only a partial, uncertain view of what is true in the world.

In [10] a model of probabilistic agents is proposed in which the main cause of uncertainty in an agent is due to its state being uncertain. There may be also other sources of uncertainty in MAS. Here we consider two of them: uncertainty of communication channels between agents of the system and uncertainty of actions. Namely, we assume that times of delivering messages through channels can be probabilistic, and some messages can be lost. Moreover, the actions can have alternatives which are executed with some probabilities. However, we assume that the choice of actions to execute at each step is deterministic, i.e. the MAS considered here are not concurrent in the sense used by M.Vardi [18].

The main result of this paper is that each such probabilistic MAS can be effectively transformed into a finite state Markov chain with polynomially computable probabilities of transitions. There is a number of papers devoted to research of complexity of verifying dynamic properties of finite state Markov chains. Our transformation of MAS to Markov chains permits to apply results of these papers to the problem of verifying behavior of different subclasses of probabilistic MAS.

Let us mention some of these papers. The research of complexity of verification problem for finite state Markov chains was initiated in the abovementioned paper by Vardi. His results on the complexity of verification of linear temporal logic (LTL) formulas on Markov chains and decision processes were improved in [7]. Analogous results for probabilistic logics of branching time (PCTL and PCTL*) were obtained in [12, 2].

The paper is organized as follows. Section 2 contains a syntactic definition of our variant of probabilistic MAS. In Section 3 we describe operational semantics of these MAS. In section 4 we present an algorithm of computing transition probabilities for Markov chains corresponding to MAS. Section 5 contains the results on the complexity of verification of probabilistic MAS obtained by applying the results of [7, 12].

2 Probabilistic MAS

There are a lot of readings and definitions of intelligent agents and multi-agent systems (see e.g. [15, 16, 21]). Here we consider the verification of behavior properties for MAS which basically conform to the so called IMPACT architecture introduced and described in detail in the book [16].

A multi-agent system $\boldsymbol{A}$ contains a finite set $\{A_1, ..., A_n\}$ of interacting intelligent agents. Any agent A has an internal database (DB) I_A consisting of a finite set of ground atoms (i.e. expressions of the form $p(c_1, \ldots, c_k)$, where p is a predicate symbol, $c_1, \ldots, c_k$ are constants; we suppose that the set of constants used by any MAS is bounded) and a message box $MsgBox_A$. Current contents of the internal DB and the message box of the agent A constitute its local state $IM_A =< I_A, MsgBox_A >$.

The agents of MAS **A** interact by sending messages of the form $msg(Sender, Receiver, Msg)$ to other agents where $Sender$ and $Receiver$ are agents (the source and the destination of the message), and Msg is a ground atom transferred.

For any pair of agents A and B in **A** there is a communication channel CH_{AB}, which receives messages sent to B by A. After some time these messages are transferred to the message box of B. We consider the length of the transfer time of the messages as a random variable identified by a discrete finite probability distribution. $p_{AB}(t)$ denotes the probability that B receives a message sent to B by A in exactly $t \geq 1$ steps after its sending (so, a constant t_0 is connected with **A** such that $p_{AB}(t) = 0$ for all A, B and $t > t_0$).

We assume that random variables for different messages are independent, and $\sum_{t=1}^{\infty} p_{AB}(t) \leq 1$. The difference $1 - \sum_{t=1}^{\infty} p_{AB}(t)$ defines the probability that the message will be lost in the channel. If $p_{AB}(1) = 1$ then any message sent to B by A will be received by the destination in the next time instant. If $p_{AB}(1) = 1$ for all agents of MAS we have synchronous variant of multi-agent systems. Such systems were considered in [8, 9]. If $p_{AB}(1) = 0.5,\ p_{AB}(2) = 0.4$ and $p_{AB}(t) = 0$ when $t > 2$, then the half of messages sent to B by A will be received in the next time, 4/10 of them will be on the path 2 steps, and average 1/10 of them will be lost in the channel.

The current state of CH_{AB} contains all the messages sent to B by A which are not received by B; they are marked by time they are in the channel. For the current state of the channel we use the same notation as for the channel, i.e. $CH_{AB} = \{(Msg, t)|$ the message Msg is in this channel during t steps of execution$\}$. For brevity we use also notations CH_{ij} and p_{ij} for $CH_{A_iA_j}$ and $p_{A_iA_j}$, respectively.

Each agent A is capable of performing a number of parameterized actions constituting its action base ACT_A. Any (parameterized) action has a name of the form $a(X_1, \ldots, X_m)$ and a set of alternatives: $a^1 =< ADD_a^1(X_1, \ldots, X_m)$, $DEL_a^1(X_1, \ldots, X_m)$, $SEND_a^1(X_1, \ldots, X_m) >, \ldots,\ a^k =< ADD_a^k(X_1, \ldots, X_m)$, $DEL_a^k(X_1, \ldots, X_m)$, $SEND_a^k(X_1, \ldots, X_m) >$. A probabilistic distribution $p_a(j)$, $1 \leq j \leq k$, is defined on these alternatives for a such that $\sum_{j=1}^{k} p_a(j) = 1$. The sets $ADD_a^j(X_1, \ldots, X_m)$ and $DEL_a^j(X_1, \ldots, X_m)$ consist of atoms of the form $p(t_1, \ldots, t_r)$, where p is r-ary predicate (for some r) in the signature of the internal DB, $t_1, \ldots, t_r$ are variables $X_1, \ldots, X_m$ or constants. These sets determine updates of the internal DB (adding and deleting facts) when the corresponding action is executed. The set $SEND_a^j(X_1, \ldots, X_m)$ consists similarly of atoms of the form $msg(A, B, p(t_1, \ldots, t_r))$, determining messages which will be sent by A to other agents. Let $c_1, \ldots, c_m$ be constants. Let us denote by $ADD_a^j(c_1, \ldots, c_m)$ the set of facts obtained by substitution of $c_1, \ldots, c_m$ instead of $X_1, \ldots, X_m$ into atoms of $ADD_a^j(X_1, \ldots, X_m)$. The sets $DEL_a^j(c_1, \ldots, c_m)$ and $SEND_a^j(c_1, \ldots, c_m)$ are defined similarly. The ground atoms $a(c_1, \ldots, c_m)$ are called *ground action names* (or simply, *ground actions*).

For example, let an agent *Accountant* works with a BD *Salary.* Then a parameterized action of salary changing can include two alternatives: the first one is to make the proposed changing, and the second - to reject them. Then these alternatives can be described in the following way:

$salary - changing^1(Name, Position, OldSum, NewSum)$:
ADD^1 ={*salary(Name, Position, NewSum)*},
DEL^1 ={*salary(Name, Position, OldSum)*},
$SEND^1$ = {*(Boss, salary_changed(Name, NewSum))*},
$salary - changing^2(Name, Position, OldSum, NewSum)$:
$ADD^2 = \emptyset$, $DEL^2 = \emptyset$,
$SEND^2$ = {*(Boss, salary_not_changed(Name))*}.
Let $p^1 = 0.8$ and $p^2 = 0.2$.

The policy of the agent A for choosing actions to execute depends on the current local state of A and is determined by a pair $< LP_A, Sel_A >$. Here LP_A is a logical program which determines a set $Perm (= Perm_{A,t})$ of ground action names permitted for execution at current time. The obligation operator Sel_A selects from $Perm$ a ground action $a(c_1, \ldots, c_q)$. We assume that Sel_A is a polynomially computable function. Then one of alternatives for the action $a(c_1, \ldots, c_q)$ (say, a^j) should be chosen with probability $p_a(j)$ to be currently executed.

This execution goes in the following way:
1) the next state of the internal base of A is obtained from the current state by deleting all the facts belonging to $DEL_a^j(c_1, \ldots, c_q)$, and then adding all the facts belonging to $ADD_a^j(c_1, \ldots, c_q)$;
2) simultaneously with changing internal DB the executing of the alternative a^j leads to changes of states of the communication channels. Namely, to any channel CH_{AB}, $B \neq A$, pairs of the form $(Ms, 0)$ are added such that $msg(A, B, Ms) \in SEND_a^j(c_1, \ldots, c_q)$.

F.e., let the *Accountant* agent has to execute the following set of actions $Obl =$ {*salary-changing*(*smith*, *engineer*, 3500, 5000),
salary-changing(*jones*, *programmer*, 4500, 6000)}.
For each of these two actions one of two alternative *salary-changing*^{1} or *salary-changing*^{2} will be chosen with probabilities 0.8 and 0.2, respectively. E.g. with probability 0.64 two first alternatives will execute. In this case after executing the facts

salary(smith, engineer, 3500), *salary(jones, programmer,4500)*
will be deleted from the internal DB, and the facts
salary(smith, engineer, 5000) and *salary(jones, programmer, 6000)*
will be added to it.
Moreover, two pairs will be placed into the channel $CH_{accountant\ boss}$:
(*salary_changed*(*smith*, 5000), 0), and *(salary_changed(jones,6000),0)*}.

To complete the definition of A and one-step semantics for it we should define LP_A, and how it does determine the current value of the set $Perm$.

As LP_A we consider logic programs with the clauses of the form

H :- L_1,...,L_n where $n \geq 0$, the head H is an action atom, the literals L_i are either action literals, or (extensional) internal DB literals, or atoms of the form

$msg(Sender, A, Msg)$ or their negations **not** $msg(Sender, A, Msg)$, or calls of some built-in polynomially computable predicates.

We suppose that the program clauses are *safe* in the sense that all variables in the head H occur *positively* in the body L_1,...,L_n, and, moreover, the program LP_A is stratified [1]. Then for any local state *state*= $< I_A,\ MsgBox_A >$ the program

$LP_{A,state} = LP_A \cup I_A \cup MsgBox_A,$

determining the set of actions which can be currently executed, is also stratified.

It is well known (see [1]) that stratified logic programs have a unique minimal model. Let $M_{A,state}$ denote such model for $LP_{A,state}$. The standard fixpoint computation procedure constructs this model in polynomial time with respect to the size of groundization $gr(LP_{A,state})$ of $LP_{A,state}$ (remember that we suppose polynomial computability of all built-in predicates). Note that the size of $gr(LP_{A,state})$ can be exponential with respect to the size of $LP_{A,state}$.

Then the set $Perm$ of actions permitted for current execution is defined as the set of ground action names contained in $M_{A,state}$. Let *Sem* denote the function defining $Perm$ from $LP_{A,state}$.

As selection operators Sel_A we permit arbitrary functions which for a given set AS of ground action names return in polynomial time some subset $Sel(AS)$ of AS. A trivial example for this is the identity function. More interesting examples are connected with defining some priority relations on actions.

3 The probabilistic MAS behavior

The global state S of the system **A** includes local states of its agents and states of all channels:

$S =< I_1, \ldots, I_n; CH_{1,2}, CH_{2,1}, \ldots, CH_{n-1,n}, CH_{n,n-1} >.$

Let $\mathbf{S}_A$ denote the set of all the global states of **A**. Then the one-step semantics of **A** defines a transition relation $S \Rightarrow_A S'$, and probabilities $p_{i,j}(t)$ induce probabilities $p(S, S')$ of these transitions.

The transition $S \Rightarrow_A S'$ starts with changes in channels and message boxes. Namely, the time counters of all the messages in channels are increased by 1, then into message box $MsgBox_j$ of any agent A_j the facts $msg(A_i, A_j, Msg)$ are placed with probability $p_{i,j}(t)$, for any Msg and i such that $(Msg, t) \in CH_{i,j}$. The pairs (Msg, t_0) can be considered as lost and are deleted from $CH_{i,j}$. After this any agent $A_i \in \mathbf{A}$ determines the set $Perm_{A_i} = Sem(LP_{A_i,state})$ of actions permitted to be currently executed, and a ground action $a_i(c_1, \ldots, c_q)$ to be executed is selected from $Perm_i$ by using the selection function Sel_{A_i}. After this an alternative a_i^j for a_i is chosen with probability $p_{a_i}(j)$, all the facts in $DEL^j_{a_i}(c_1, \ldots, c_q)$ are deleted from I_i, and all the facts in $ADD^j_{a_i}(c_1, \ldots, c_q))$ are added to it. Moreover, the communication channels $CH_{i,m}$ are complemented by entries $(ms, 0)$ such that messages $msg(A_i, A_m, ms)$ are in $SEND^j_{a_i}(c_1, \ldots, c_q))$. The message boxes of all the agents are emptied (in fact this does not restrict generality since all needed data can be transferred before from message boxes into internal DBs).

So, the transition S $\Rightarrow_A$ S' is computed by the following probabilistic algorithm:

A-step (**Input:** S ; **Output:** S')

(1)**FOR EACH** $A_i, A_j \in \mathbf{A}$ $(i \neq j)$ **DO**
(2) **FOR EACH** $(Msg, t) \in CH_{i,j}$ **DO**
(3) **BEGIN** $CH_{i,j} := (CH_{i,j} \setminus \{(Msg, t)\});$
(4) **IF** $t \leq t_0$ **THEN** $CH_{i,j} := (CH_{i,j} \cup \{(Msg, t+1)\}$ **END**;
(5)**FOR EACH** $A_i, A_j \in \mathbf{A}(i \neq j)$ **DO**
(6) **FOR EACH** $(Msg, t) \in CH_{i,j}$ **DO** with probability $p_{i,j}(t)$
(7) **BEGIN** $CH_{i,j} := (CH_{i,j} \setminus \{(Msg, t)\});$
(8) $MsgBox_j := MsgBox_j \cup \{msg(A_i, A_j, Msg)\}$
(9) **END**;
(10) **FOR EACH** $A_i \in \mathbf{A}$ **DO**
(11) **BEGIN** $Perm_i := Sem(LP_{A_i,state});$
(12) Let $Sel_{A_i}(Perm_i)$ be $a_i(c_1, \ldots, c_q);$
(13) Let $a_i^1, \ldots, a_i^k$ be all the alternatives of a_i;
(14) Let us choose an alternative $a_i^j, 1 \leq j \leq k$, for a_i with probability $p_{a_i}(j)$;
(15) $I_i' := ((I_i \setminus DEL_{a_i}^j(c_1, \ldots, c_q)) \cup ADD_{a_i}^j(c_1, \ldots, c_q));$
(16) **FOR EACH** $(m \neq i)$ **DO**
(17) $CH_{i,m}' := (CH_{i,m} \cup \{(ms, 0) | msg(A_i, A_m, ms) \in SEND_{a_i}^j(c_1, \ldots, c_q)\});$
(18) $MsgBox_i := \emptyset;$
(19) **END;**
(20) **RETURN** S'.

This definition of semantics for MAS permits to connect a finite Markov chain **MC(A)** with any MAS **A**. The states of **MC(A)** are global states from S_A, and probabilities $p_A(S, S')$ of transitions from S to S' can be computed by the algorithm described in the next section. The behavior of **A** for an initial global state S^0 is described by a tree $t_A(S^0)$ of possible trajectories of this chain with root labelled by S^0. Nodes of this tree are labelled by global states of **A,** and from any node on the level t labelled by S goes an edge labelled by $p_A(S, S')$ to a node labelled by S' if $p_A(S, S') > 0$.

Note that the cardinality of the set of states of the Markov chain **MC(A)** is exponential with respect to the size of **A** in the worse case, if **A** is ground, and even double exponential if **A** is non-ground.

4 Probabilistic MAS as Finite Markov Chains

We note that all the stochastics in the program **A-**step is concentrated in lines 6-9 and 14, which determine as messages transfer to message boxes with accord to probabilities $p_{i,j}(t)$ and which alternative of the action a_i is chosen. We assume that all the probabilistic choices in these lines are independent.

The following effective procedure permits to compute the probability $p_{\mathbf{A}}(S, S')$ of transition $S \Rightarrow_A S'$:

Algorithm *Prob(S, S')*

(1) **FOR EACH** $A_i, A_j \in \mathbf{A}$ $(i \neq j)$ **DO**
(2) **BEGIN** $M[i,j] := \{(m,t) | ((m,t) \in CH_{i,j})$ & $((m, t+1) \notin CH'_{i,j})\}$;
(3) $p_{i,j} := \prod\{p_{i,j}(t) | ((m,t) \in M[i,j]\}$;
(4) **END**;
(5) **FOR EACH** $A_j \in \mathbf{A}$ **DO**
(6) **BEGIN** $MsgBox_j := \emptyset$;
(7) **FOR EACH** $A_i \in \mathbf{A}(i \neq j)$ **DO**
(8) $MsgBox_j := MsgBox_j \cup \{msg(A_i, A_j, m) | \exists t((m,t) \in M[i,j])\}$
(9) **END**;
(10) **FOR EACH** $A_i \in$ **A** **DO**
(11) **BEGIN** $Perm_i := Sem(LP_{A_i,state})$;
(12) Let $Sel_{A_i}(Perm_i)$ be $a_i(c_1, \dots, c_q)$;
(13) Let $a_i^1, \dots, a_i^k$ be all the alternatives of a_i;
(14) $p_i := \sum_j \{p_{a_i}(j) | I'_i = ((I_i \setminus DEL^j_{a_i}(c_1, \dots, c_q)) \cup ADD^j_{a_i}(c_1, \dots, c_q))$ **and** $(\bigwedge_{m \neq i} \{ms | (ms, 0) \in CH'_{i,m}\} = \{ms | msg(A_i, A_m, ms) \in SEND^j_{a_i}(c_1, \dots, c_q))\}\}$;
(15) **END**;
(16) $p_{\mathbf{A}}(S, S') := \prod\{p_{i,j} | 1 \leq i, j \leq n; j \neq i\} * \prod_{i=1}^n p_i$;
(17) **RETURN** $p_{\mathbf{A}}(S, S')$.

Note that $M[i,j]$ in the line 2 is the set of entries of the channel $CH_{i,j}$ which are put into $MsgBox_{A_j}$. Then $p_{i,j}$ in the line 3 is the probability of the event: the set of messages from A_i to A_j included into $MsgBox_{A_j}$ is equal to $M[i,j]$. Moreover, p_i in the line 14 is the probability to obtain a new internal state I'_i from I_i after applying the action $a_i(c_1, \dots, c_q)$.

Theorem 1 *The algorithm **Prob(S, S')** computes probability $p(S, S')$ of transition $S \Rightarrow_A S'$ in time polynomial on sum of sizes of MAS $\boldsymbol{A}$ and states S and S' , i.e. on $|\boldsymbol{A}| + |S| + |S'|$ (we include into the size $|\boldsymbol{A}|$ of MAS the sizes of all signatures, of the set of constants, of the agent descriptions with their action bases and groundizations of agents' programs and of probabilty disributions for action alternatives and communication channels).*

5 Complexity of Veryfying Dynamic Properties of MAS

Traditionally behavior properties of discrete dynamic systems are specified in some variants of temporal logics, see e.g [6]. There are two basic types of such logics: of linear time and of branching time. Normally, states of Markov chains are considered as non-structured. So, dynamic properties of such Markov chains can be adequately represented by formulas of propositional versions of these logics. States of MAS have a structure of finite models. Hence it is natural to extend logics for specifying their dynamic properties by introducing first-order features (as in [8, 9]. Namely, the extension is that ordinary closed first-order formulas in signature of internal databases of agents (called *basic state formulas*) can be used in formulas instead of propositional variables. Then the possibillity of transferring results on complexity of verifying finite Markov chains to probabilistic MAS stems from the well-known fact that the basic state formulas can be verified on finite models of states in polynomial space (or even in poynomial time for formulas of bounded quantifier depth)..

The problem of verification of dynamic properties for logics of linear and branching time are formulated in a somewhat different way.
- *Linear time*: for a given probabilistic MAS $\mathbf{A}$, its initial state S^0 and formula F of FLTL describing a property of trajectories to find the measure (probability) $p_A(S^0, F)$ of the set of trajectories of the tree $t_A(S^0)$ which satisfy F. If this probability is equal to 1 we say that the pair $(\mathbf{A}, S^0)$ satisfies F.
- *Branching time*: A main role in branching time logics play formulas expressing properties of states (not trajectories). The measure of satisfiability of such formulas does not express stochastic properties of behavior of the system. Because of this it was proposed in [12]: to replace in formulas quantifiers on trajectories by probability bounds. E.g. formula $[Gf]_{>p}$ means that the measure of trajectories starting in the current state with all their states satisfying f is greater than p. The logic obtained is called PCTL.

Now we can state some of numerous results on complexity of verifying dynamic properties of MAS which can be obtained by transferring corresponding results from Markov chains.

- *Linear time* In this case we can apply Theorem 3.1.2.1 of the paper [7]. This theorem states existence of two algorithms: 1) testing if a given finite Markov chain $\mathbf{M}$ satisfies a formula F of PLTL in time $O(|M|2^{|F|})$, or in space polynomial in $|F|$ and polylogarithmic in $|M|$, and 2) computing the probability $p_M(F)$ of satisfaction F on $\mathbf{M}$ in time exponential in $|F\}$ and polynomial in $|M|$.

To apply this theorem we need only to use Theorem 1 and the remark from the end of the section 3 on estimates of the size of $\mathbf{MC(A)}$ with respect to the size of **A.** We give here only few of corollaries.

Theorem 2 *(1) There exists an algorithm which checks satisfiability of a formula F from FLTL in a state S of a ground probabilistic MAS* $\boldsymbol{A}$ *in polynomial space on* $|\boldsymbol{A}|$ *and* $|F|$.

(2)There exists an algorithm which computes probability $p_A(S^0, F)$ *for any ground probabilistic MAS* $\boldsymbol{A}$ *and formula F in time exponential both in* $|\boldsymbol{A}|$ *and* $|F|$.

(3) There exists an algorithm which computes probability $p_A(S^0, F)$ *for any (non-ground) probabilistic MAS* $\boldsymbol{A}$ *and formula F in time exponential in* $|F|$ *and double exponential in* $|\boldsymbol{A}|$.

- *Branching time*: In [12] an algorithm is constructed which decides whether a formula F of PCTL is satisfied in a Markov chain M. The time complexity of this algorithm is $O(|M|^3 * |F|$. From this we obtain (using the first-order extension FPCTL instead of PCTL)

Theorem 3 *(1) There exists an algorithm which checks satisfiability of a formula F from FPCTL in a state S of a ground probabilistic MAS* $\boldsymbol{A}$ *in exponential time on* $|\boldsymbol{A}|$ *and linear time on* $|F|$.

(2)There exists an algorithm which checks satisfiability of a formula F from FPCTL in a state S of a (non-ground) probabilistic MAS $\boldsymbol{A}$ *in time double exponential on* $|\boldsymbol{A}|$ *and linear on* $|F|$.

We note that the estimates for $|\mathbf{MC(A)}|$ above were given for worse case. However, in many cases these estimates can be drastically decreased (from exponential to polynomial or from double exponential to exponential). E.g., if arities of predicates in internal DBs, action bases and messages are bounded, then the cardinality of set of global states for nonground MAS is bounded by some exponential of a polynomial. So, in the assertion (3) of Theorem 2 words "double exponential in $|\mathbf{A}|$" can be replaced by "exponential of a polynomial of $|\mathbf{A}|$". Moreover, it may happen under constructing $\mathbf{MC(A)}$ that many global states of **A** are not reachable or not acceptable. This can also lead to a serious decreasing of complexity of problem of verification.

6 Conclusion

In this paper we showed how probabilistic multi-agent systems can be transformed to finite state Markov chains. This permitted to obtain some results on complexity of verifying dynamic properties of MAS by applying corresponding results for finite Markov chains known from the literature. Note that we considered here only MAS with deterministic selection of actions. Of course, it is also interesting to consider verification problem for MAS with non-deterministic selection of actions. It seems that in this case results on verifying concurrent Markov chains (Markov decision processes) [18, 7, 13] can be applied.

References

1. Apt K. R., Logic programming. In: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science. Volume B. Formal Models and Semantics, Chapter 10, Elsevier Science Publishers B.V. 1990, 493-574.
2. Aziz A., Singhal V., Balarin F., and Brayton R.K. It usually works: The temporal logic of stochastic systems. Lecture Notes in Computer Science, 939, 1995, 155-165.
3. Barringer, H., Fisher, M., Gabbay, D., Gough, G., and Owens R. METATEM: An Introduction. *Formal Aspects of Computing,* 1995, 7:533-549.
4. Bordini R., FisherM., Pardavila C., and Wooldridge M. Model checking AgentSpeak. AAMAS 2003, pp. 409416, 2003.
5. Benerecetti, M., Guinchiglia, F., and Serafini, L. Model checking multiagent systems. *Technical Report # 9708-07.* Instituto Trentino di Cultura, 1998.
6. Clarke E.M., Grumberg O. and Peled D., Model checking, MIT Press, 2000.

7. Courcoubetis C., Yannakakis M., The complexity of probabilistic verification. J. ACM, v. 42, 4, 1995, 857-907.
8. Dekhtyar M., Dikovsky A., and Valiev M., On feasible cases of checking multi-agent Systems Behavior. Theoretical Computer Science, Elsievier Science, 2003, vol. 303, no. 1, 63-81.
9. Dekhtyar M.I., Dikovsky A.Ja., Valiev M.K., On complexity of verification of interacting agents' behavior. Annals of Pure and Applied Logic, 141, 2006, 336 – 362.
10. Dix J., Nanni M., and Subrahmanian V. S. , Probabilistic agent reasoning. ACM Transactions of Computational Logic, 1(2):201-245, 2000.
11. Giordano L., Martelli A., and Schwind C. Verifying communication agents by model checking in a temporal action Logic. JELIA 2004, LNAI 3229, 57-69.
12. Hansson H., Jonsson B. A logic for reasning about time and reliability. Formal Aspects of Computing, 6(5), 1994, 512-535.
13. Kwiatkowska Marta. Model Checking for probability and time: from theory to practice. In: Proc. 18th IEEE Symposium on Logic in Computer Science, 2003, 351-360.
14. Rao A.S., Georgeff M.P., Modeling rational agents within a BDI architecture. Proc 2nd Intern. Conf. on Principles of Knowledge Representation and Reasoning, Morgan Kaufman Publishers, 1991.
15. Shoham, Y., Agent oriented programming. *Artificial Intelligence*, 1993, 60:51-92.
16. Subrahmanian V. S., Bonatti P., Dix J., et al., Heterogeneous agent systems, MIT LPess, 2000.
17. van der Hoek W., Wooldridge M., Tractable multiagent planning for epistemic goals. AAMAS'02, Bologna, Italy, 2002.
18. Vardi M.Y., Automatic verification of probabilistic concurrent finite state programs. In: Proceedings of 26th IEEE Symposium on Foundations of Computer Science. IEEE, New York, 327-338.
19. Wooldridge, M., Dunne, P.E., The Computational complexity of Agent Verification. In: *J.-J. Meyer and M. Tambe (eds.), Intelligent Agents VIII. Springer-Verlag Lecture Notes in AI Volume,* March 2002.
20. M. Wooldridge, M. Fisher, M.-P. Huget, and S. Parsons. Model Checking Multiagent systems with MABLE. In: *Proc. of the First Intern. Conf. on Autonomous Agents and Multiagent Systems (AAMAS-02)*, Bologna, Italy, July 2002.
21. Wooldridge, M., Jennings N. Intelligent agents: Theory and practice. *The Knowledge Engineering Review,* 1995, 10(2).