СЛОЖНОСТЬ ВЕРИФИКАЦИИ МУЛЬТИАГЕНТНЫХ СИСТЕМ С ВЕРОЯТНОСТНЫМИ СОСТОЯНИЯМИ И ПРОГРАММАМИ[1]

Валиев М.К., *к.ф.-м.н.*
Институт прикладной математики им. М.В.Келдыша РАН
(495)250-79-38, valiev@keldysh.ru
Дехтярь М.И., *к.ф.-м.н., доцент*
Тверской государственный университет
(482-2)365410, Michael.Dekhtyar@tversu.ru

1. ВВЕДЕНИЕ

В этой работе мы продолжаем изучение сложности верификации динамических свойств мультиагентных систем (МАС), состоящих из вероятностных интеллектуальных агентов, которое было начато в наших предыдущих работах [5,10,11]. В указанных работах вероятностными были каналы связи между агентами и действия. При этом предполагалось, что агенты действуют, имея точную информацию о своих состояниях (базах фактов), и для определения своих действий используют логические программы обычного типа. Здесь мы считаем, что состояния агентов также являются вероятностными, а выбор действий определяется вероятностными логическими программами. Мы определяем операционную семантику для таких обобщенных вероятностных МАС и обобщаем на них конструкцию из вышеуказанных работ, которая по вероятностной МАС строит конечную марковскую цепь, моделирующую ее работу. Это позволяет применить (так же, как в [5,10,11]) к рассматриваемым здесь обобщенным вероятностным МАС алгоритмы верификации конечных Марковских цепей из [2, 7].

.

2. ВЕРОЯТНОСТНЫЕ МАС

Имеются различные подходы к определению интеллектуальных агентов (см., например, [9,13]). Наше определение вероятностного агента близко к определению, рассмотренному в [6,9].

---

[1] **Эта работа выполнена при поддержке РФФИ (гранты 07-01-00637-а и 08-01-00241-а).**

Вероятностная мультиагентная система (МАС) $A = \{A_1,...,A_n\}$ состоит из конечного множества $\{A_1,...,A_n\}$ взаимодействующих вероятностных интеллектуальных агентов A_i. У каждого агента A системы имеется *внутренняя вероятностная база данных* (ВБД) I_A, содержащая конечное множество аннотированных базисных (ground) атомов вида $q(c_1,\dots, c_k)$:p, где *q*-предикатный символ, $c_1,\dots, c_k$ – константы, $p\in[0,1]$ – степень уверенности в факте $q(c_1,\dots, c_k)$ (или вероятность этого факта). Отметим, что множество используемых данной системой констант ограничено. Кроме ВБД у агента A имеется почтовый ящик $MsgBox_A$, в котором находятся сообщения, полученные им перед текущим шагом от других агентов системы. Текущие содержимые внутренней ВБД и почтового ящика агента A составляют его текущее локальное состояние $IM_A=<I_A,MsgBox_A>$.

Агенты из *A* общаются между собой посредством передачи сообщений вида msg(Sender, Receiver, Msg), где Sender и Receiver – имена агентов (источника и адресата), а Msg - (передаваемый) базисный атом.

Для каждой пары агентов A, B из *A* имеется канал связи CH_{AB}, в который попадают сообщения, посылаемые агентом A агенту B. Затем из этого канала они попадают в почтовый ящик $MsgBox_B$. Время пребывания каждого сообщения «в пути» мы будем рассматривать как случайную величину, задаваемую конечным дискретным распределением вероятностей. Через $p_{AB}(t)$ обозначим вероятность того, что B получит сообщение, посланное ему агентом A, ровно через $t \geq 1$ шагов (тактов) после его отсылки (t0 будет обозначать минимальное число такое, что $p_{AB}(t) =0$ для всех $t > t0$ и всех агентов A и B системы).

Для разных сообщений соответствующие случайные величины будем считать независимыми. Мы предполагаем, что $\sum_{t=1}^{\infty} p_{AB}(t) \leq 1$. Тогда разность $1 - \sum_{t=1}^{\infty} p_{AB}(t)$ определяет вероятность того, что сообщение никогда не достигнет адресата, т.е. будет утеряно в канале. Текущее состояние канала CH_{AB} будет включать все сообщения, посланные агентом A агенту B, которые еще не дошли до B, с указанием времени их нахождения в канале. Мы будем обозначать текущее состояние канала так же как и сам канал, т.е.

$CH_{AB} =\{(Msg, t)$ | *сообщение Msg от агента A агенту b находится в канале t тактов*$\}$.

Мы будем также использовать сокращения CH_{ij} и p_{ij} для CH_{AiAj} и p_{AiAj}, соответственно.

С каждым агентом A связана его база ACT_A *параметризованных действий* вида $<a(X_1,\dots,X_m), PUT_a(X_1,\dots,X_m), SEND_a(X_1,\dots,X_m)>$. Здесь

$a(X_1,\ldots,X_m)$ – (параметризованное) имя действия, $PUT_a(X_1,\ldots,X_m)$ – список аннотированных атомов вида $q(t_1,\ldots,t_k)$:p, где q – k-местный предикат из сигнатуры внутренней ВБД, $t_1,\ldots,t_k$ – либо константы, либо параметры $X_1,\ldots,X_m$, p – вероятность атома $q(t_1,\ldots,t_k)$. Это множество определяет изменения внутренней ВБД при выполнении данного действия (это будет уточнено в следующем разделе). Список $SEND_a(X_1,\ldots,X_m)$ содержит сообщения вида msg(A,B, $p(t_1,\ldots,t_k)$), отправляемые другим агентам. Пусть $c_1,\ldots,c_m$ – константы. Обозначим через $PUT_a(c_1,\ldots,c_m)$ множество базисных аннотированных фактов, получаемых подстановкой $c_1,\ldots,c_m$ вместо $X_1,\ldots,X_m$ в атомы из $PUT_a(X_1,\ldots,X_m)$. Аналогично определяется и $SEND_a(c_1,\ldots,c_m)$. Базисные атомы вида $a(c_1,\ldots, c_l)$ назовем базисными именами действий (или просто базисными действиями).

Конкретный выбор действий агента, *возможных* в данном локальном состоянии, определяется его *вероятностной логической программой* LP_A. В качестве программ LP_A мы рассматриваем вероятностные логические программы с предложениями вида

$H{:}p :- L_1,\ldots,L_n$.

Здесь H – атом действия, т.е. имеет вид $a(t_1,\ldots,t_m)$, где $t_1,\ldots,t_m$ – либо константы, либо переменные, $p\in[0,1]$; литералы L_i – либо аннотированные атомы действий, либо (экстенсиональные) аннотированные атомы вида $q(t_1,\ldots,t_k)$:[l,u] с предикатами q из сигнатуры внутренней БД и аннотациями [l,u] – подинтервалами отрезка [0,1], либо литералы сообщений вида msg(Sender, A, Msg) или not msg(Sender, A, Msg), либо атомы с сигнатурой из некоторых вычислимых в полиномиальное время встроенных предикатов.

Такие вероятностные логические программы являются частным случаем логических программ с интервальными вероятностями, определенных в работе [8]. В [3] было показано, что в общем случае программы из [8] не имеют естественной просто вычислимой семантики. Можно показать, что для определенного выше варианта вероятностных логических программ такая семантика существует и может быть вычислена за полиномиальное время от размера базисной развертки $gr(LP_{A,state})$ программы $LP_{A,state} = LP_A \cup I_A \cup MsgBox_A$. Обозначим через $Perm_A$ (= $Sem(LP_{Ai,state})$) множество базисных аннотированных имен действий $\{a_1(c_{1,1},\ldots c_{m1,1}){:}p_1, \ldots , a_k(c_{1,k},\ldots c_{mk,k}){:}p_k\}$ определяемых семантикой программы $gr(LP_{A,state})$.

Заметим, что введенные в этой работе вероятностные МАС можно считать обобщением рассмотренных нами ранее систем. А именно, действие по удалению факта из внутренней базы можно моделировать заменой вероятности этого факта на нулевую, а вероятностные действия можно моделировать соответствующим усложнением используемых логических программ.

3. ПОВЕДЕНИЕ ВЕРОЯТНОСТНЫХ МАС

Определим операционную семантику введенных в предыдущем разделе вероятностных МАС.

Глобальное состояние S системы **A** включает в себя локальные состояния ее агентов и состояния всех ее $(n^2\text{-}n)$ каналов:

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

Обозначим через $\mathbf{S_A}$ множество всех глобальных состояний МАС **A**. Тогда *одношаговая семантика* МАС **A** задает отношение $S \Rightarrow_{\mathbf{A}} S'$ перехода (за один шаг) на множестве $\mathbf{S_A}$, а вероятности, участвующие в определениях агентов **A,** индуцируют вероятности таких переходов p(S, S').

Переход $S \Rightarrow_A S'$ начинается с работы каналов и формирования нового содержимого почтовых ящиков. Сначала каждый канал увеличивает на 1 счетчик времени у всех находящихся в нем сообщений. Пары (Msg, t) такие, что $t>t_0$, удаляются из $CH_{i,j}$. Затем для каждой пары (Msg, t) $\in CH_{i,j}$ в почтовый ящик $MsgBox_j$ агента A_j с вероятностью $p_{i,j}(t)$ помещается факт msg(A_i, A_j ,Msg). После этого каждый агент $A_i \in$ **A** формирует множество всех допустимых на данном шаге аннотированных базисных действий $Perm_i = Sem(LP_{i,state})$. Затем по $Perm_i$ формируется множество выполняемых агентом A_i действий Obl_i: для каждого аннотированного атома $a(c_1,\ldots c_m)$:p из $Perm_i$ действие $a(c_1,\ldots c_m)$ помещается в Obl_i с вероятностью p. Почтовые ящики всех агентов МАС **A** после этого опустошаются, т.е. полученные сообщения "забываются". Разумеется, это не ограничивает общности, поскольку агент может все нужные ему данные перенести из почтового ящика в свою базу данных. После этого каждый агент A_i выполняет действия из Obl_i следующим образом. Обозначим через UPD_i множество $\{q(t_1,\ldots,t_k)\text{:}p \mid p = \max\{ p' \mid q(t_1,\ldots,t_k)\text{:}p' \in PUT_a(c_1,\ldots,c_m)$ *для некоторого* $a(c_1,\ldots c_m)$ из $Obl_i \}\}$**.** Тогда новое состояние ВБД I_i получается путем удаления из I_i всех старых фактов из множества $UPD\_OLD_i = \{q(t_1,\ldots,t_k)\text{:}r \mid$ *для некоторого* p $q(t_1,\ldots,t_k)\text{:}p \in UPD_i\}$ и добавления к I_i новых аннотированных фактов из UPD_i.[2]. И наконец,

---

[2] **Заметим, что использование функции max в определении UPD соответствует оптимистическому взгляду на изменение вероятностей фактов. Можно было бы выбрать и другие способы модификации вероятностей (например, вместо max использовать min).**

агент A_j добавляет в каждый канал CH_{ij} $(i \neq j)$ все пары вида (Msg, 0), где Msg является базисным экземпляром некоторого сообщения вида $msg(A_i, A_j, p(t_1,\ldots,t_k))$ из множества $SEND_a(c_1,\ldots c_m)$ для некоторого $a(c_1,\ldots c_m)$ из Obl_i.

Таким образом, переход $S \Rightarrow_A S'$ вычисляется следующим вероятностным алгоритмом:

А-шаг (Вход: S **;** Выход: **S')**

(1) FOR EACH $A_i, A_j \in A$ $(i \neq j)$ 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 $CH_{i,j} := (CH_{i,j} \setminus \{(Msg, t+1)\})$; END

(5) FOR EACH $A_i, A_j \in A$ $(i \neq j)$ DO

(6) FOR EACH $(Msg, t) \in CH_{i,j}$ DO с вероятностью $p_{i,j}(t)$

(7) BEGIN $CH_{i,j} := (CH_{i,j} \setminus \{(Msg, t)\})$;

(8) $MsbBox_j := MsbBox_j \cup \{msg(A_i, A_j, Msg)\}$

(9) END;

(10) FOR EACH $A_i \in A$ DO

(11) BEGIN $Perm_i := Sem(LP_{Ai,state})$;

(12) FOR EACH $a(c_1,\ldots c_m):p \in Perm_i$ DO

(13) поместить $a(c_1,\ldots c_m)$ в Obl_i с вероятностью p;

(14) $UPD_i := \{ q(t_1,\ldots,t_k):p \mid p = \max\{ p' \mid$
$q(t_1,\ldots,t_k):p' \in PUT_a(c_1,\ldots,c_m) \wedge a(c_1,\ldots c_m) \in Obl_i \}\}$;

(15) $UPD\_OLD_i = \{q(t_1,\ldots,t_k):r \in I_i \mid q(t_1,\ldots,t_k):p \in UPD_i\}$;

(16) $I_i' := ((I_i \setminus UPD\_OLD_i) \cup UPD_i$;

(17) FOR EACH $(m \neq i)$ DO

(18) $CH_{i,m}' := (CH_{i,m} \cup$
$\{(ms,0) \mid msg(A_i, A_m, Ms) \in SEND_a(c_1,\ldots,c_k)$
$\wedge a(c_1,\ldots c_k) \in Obl_i \})$;

(19) $MsgBox_i := \varnothing$;

(20) END;

(21) RETURN $S'=(I_i',\ldots ,I_n', CH_{1,2}',\ldots, CH_{n-1,n}')$.

Для завершения определения одношаговой семантики **А** нужно еще дать точное описание для $Sem(LP_{Ai,state})$, чему посвящен следующий раздел.

4. ВЫЧИСЛЕНИЕ СЕМАНТИКИ ВЕРОЯТНОСТНОЙ ЛОГИЧЕСКОЙ ПРОГРАММЫ

В этом пункте мы рассмотрим вычисление оператора Sem(P) для базисной вероятностной логической программы P. Обозначим множество всех базисных (неаннотированных) атомов (эрбранов универсум, включающий как экстенсиональные атомы, так и атомы действий) через **U.** Интерпретация f: U →[0,1] сопоставляет каждому атому $q(c_1,\ldots c_m)\in\mathbf{U}$ его вероятность $f(q(c_1,\ldots c_m))$. Атом действия $a(c_1,\ldots c_m){:}p$ *выполнен на интерпретации* f, если $p \geq f(a(c_1,\ldots c_m))$. Аннотированный экстенсиональный атом вида $q(t_1,\ldots,t_k){:}[l,u]$ *выполнен на интерпретации* f, если $l \leq f(q(t_1,\ldots,t_k)) \leq u$. Выполнимость литералов сообщений вида msg(Sender, A, Msg) или not msg(Sender, A, Msg) определяется относительно текущего состояния $MsgBox_A$ обычным образом. Выполнимость встроенных предикатов определяется их естественной семантикой. Предложение $a(c_1,\ldots c_m){:}p$:- $L_1,\ldots,L_n$ *выполняется на интерпретации* f (для данного $MsgBox_A$), если при условии, что каждый L_i выполняется на f, имеет место неравенство $f(a(c_1,\ldots c_m)) \geq p$. Интерпретация f является *моделью* P, если на ней выполнены все предложения P. Определим на множестве интерпретаций частичный порядок $\leq$ следующим образом: $f1 \leq f2 \Leftrightarrow$ для каждого атома $q \in\mathbf{U}$ $f1(q) \leq f2(q)$. Модель f программы P назовем *минимальной*, если для всякой другой модели f1 программы P неверно, что $f1 \leq f$. Множество моделей P замкнуто относительно «минимизации».

Лемма 1. Пусть f1 и f2 – модели программы P. Тогда и интерпретация $f = \min(f1,f2)= \{\, q{:}p \mid q \in\mathbf{U} \wedge p=\min(f1(q), f2(q))\}$ является моделью P.

Из этой леммы следует существование минимальной модели P. Ее вычисление обеспечивается процедурой вычисления неподвижной точки, аналогичной процедуре из [8].

Теорема 1. Для всякой вероятностной логической программы P (определенного в этой работе типа) существует минимальная модель $fmin_P$, которая вычислима за полиномиальное время от gr(P).

Определим Perm = Sem(P) как множество всех аннотированных атомов действий $a(c_1,\ldots c_m){:}p$ из $fmin_P$.

5.ВЫЧИСЛЕНИЕ ВЕРОЯТНОСТЕЙ ПЕРЕХОДОВ

В соответствии с вышеприведенным определением семантики с МАС **A** можно связать Марковскую цепь MC(**A**) с множеством состояний $\mathbf{S}_A$ и вероятностями переходов $p_{\mathbf{A}}(S, S')$ между ними. Поведение ***A*** в начальном глобальном состоянии S^0 описывается деревом $T_A(S^0)$ возможных траекторий этой цепи, начинающихся с S^0. Узлы этого дерева помечены глобальными состояниями системы, причем каждый узел, находящийся на (t+1)-ом уровне и помеченный состоянием S', связан с узлом на t-ом уровне с пометкой S такой, что возможен впереход $S \Rightarrow_A S'$ с некоторой положительной вероятностью $p_{\mathbf{A}}(S, S')$. В этом разделе мы опишем алгоритм вычисления $p_{\mathbf{A}}(S, S')$ и оценим его сложность относительно размера системы **A**.

Заметим, что количество состояний цепи MC(**A**) в худшем случае может быть экспоненциальной относительно размера **A** , если **A** - базисная, и даже двойной экспоненциальной, если **A** - не базисная (в размер |**A**| МАС **A** входят размеры всех сигнатур, множества констант, описаний агентов, включающих их базы действий и базисные развертки программ агентов, и распределений вероятностей).

Отметим, что источниками неопределенности в алгоритме **А-шаг** являются операторы в строках 5-9 и 13, которые определяют, как сообщения попадают в почтовые ящики агентов с учетом вероятностей времен их пересылки и как выбираются действия агента, выполняемые на текущем шаге. Мы предполагаем, что все вероятностные выборки независимы.

Это позволяет предложить следующую эффективную процедуру вычисления вероятности p(S, S') перехода $S \Rightarrow_A S'$:

Алгоритм ***Prob(S, S')***

(1) FOR EACH $A_i, A_j \in A$ $(i \neq j)$ DO

(2) BEGIN $M[i,j] := \{(m, t) \mid ((m, t) \in CH_{i,j})$ & $((m,t+1) \notin CH'_{i,j})\}$;

(3) $p_{i,j} := \prod \{ p_{i,j}(t) \mid (m, t) \in M[i,j]\}$

(4) END;

(5) FOR EACH $A_j \in A$ DO

(6) BEGIN $MsgBox_j := \varnothing$;

(7) FOR EACH $A_i \in A$ $(i \neq j)$ DO

(8) $MsgBox_j := MsgBox_j \cup \{msg(A_i, A_j, m) \mid \exists t\, ((m,t) \in M[i,j])\}$

(9) END;

(10) FOR EACH $A_j \in A$ DO

(11) BEGIN $Perm_i := Sem(LP_{A,state})$;

(12) $PermAct$:- $\{ a(c_1,\ldots c_m) \mid a(c_1,\ldots c_m){:}p \in Perm_i\}$;

(12) $p_{i,} := 0$;

(13) FOR EACH Obl, где Obl - подмножество из PermAct DO

(14) BEGIN $UPD := \{ q(t_1,\ldots,t_k){:}p \mid p = \max\{ p' \mid$
$q(t_1,\ldots,t_k){:}p' \in PUT_a(c_1,\ldots,c_m) \wedge a(c_1,\ldots c_m) \in Obl \}\}$;

(15) UPD_OLD= $\{q(t_1,\ldots,t_k){:}r\in I_i \mid q(t_1,\ldots,t_k){:}p \in UPD\}$;

(16) IF $(I_i' := ((I_i \setminus UPD\_OLD) \cup UPD) \land (\land_{m\neq i} \{ms \mid (ms,0) \in CH_{i,m}'\} = \{ms \mid msg(Ai,Am,ms)\in SEND_a(c1,\ldots,cq)$ *для некоторого* $a(c1,\ldots cm)\in Obl\})$

(17) THEN $p_i := p_i + \prod \{p_a \mid a(c_1,\ldots c_m)\in Obl \land a(c_1,\ldots c_m){:}p \in Perm_i\}$

(18) END;

(16) $p(S, S') := \prod \{ p_{i,j} \mid 1 \le i, j \le n, j \neq i \} * \prod \{ p_i \mid 1 \le i \le n\}$;

(18) RETURN $p(S, S')$.

Теорема 2. Алгоритм *Prob(S, S')* вычисляет вероятность $p(S, S')$ перехода $S \Rightarrow_A S'$. Время работы *Prob(S, S')* ограничено величиной $2^r pol(|A| + |S| + |S'|)$, где r – максимальное число различных базисных действий одного агента системы, *pol* – некоторый полином, а $|A| + |S| + |S'|$ - сумма размеров МАС **A** и размеров исходного и результирующего состояний S и S'.

Заметим, что экспонента 2^r возникла из-за перебора всех подмножеств множества PermAct в строке (13) алгоритма. Если ограничить выбор действий для выполнения фиксированным числом, например, самых вероятных действий, то алгоритм становится полиномиальной сложности.

5. ВЕРИФИКАЦИЯ ДИНАМИЧЕСКИХ СВОЙСТВ

В предыдущих работах [5,10,11] нами было показано, как известные результаты [2, 7] о верификации свойств конечных Марковских цепей могут быть использованы для получения оценок сложности верификации для некоторых моделей вероятностных МАС. При этом основой для рассуждений была теорема 1 из [11], обобщением которой является теорема 2 из предыдущего раздела. Аналогичные рассуждения с привлечением теоремы 2 позволяют использовать те же результаты из [2,7] для распространения оценок из [11] на введенные в этой работе обобщенные вероятностные МАС. При этом так же как и в [11] мы рассматриваем верификацию динамических свойств, задаваемых формулами некоторых вариантов FLTL и FPCTL предикатной логики линейного и ветвящегося времени (об использовании временных логик для верификации динамических свойств (model checking) см., например, [1,12]).

Так же как и в [11] мы приведем только некоторые из следствий применения результатов из [2,7] к (обобщенным) вероятностным МАС.

Теорема 3. (1) Существует алгоритм, который проверяет выполнимость FLTL-формулы F на состоянии S базисной вероятностной МАС **A** в памяти, полиномиальной от |**A**| и | F|.

(2) Существует алгоритм, вычисляющий вероятность $p_{\mathbf{A}}(S^0, F)$ для базисной вероятностной МАС **A** и формулы F за время, экспоненциально зависящее от |**A**| и |F|.

(3) Существует алгоритм, вычисляющий вероятность $p_{\mathbf{A}}(S^0, F)$ для любой (небазисной) вероятностной МАС **A** и формулы F за время, дважды экспоненциально зависящее от |**A**| и экспоненциальное от | F|.

Теорема 4. (1) Существует алгоритм, который проверяет выполнимость FPCTL-формулы F на состоянии S базисной вероятностной МАС A за время, зависящее экспоненциально от |A| и линейно от |F|.

(2)Существует алгоритм, проверяющий выполнимость FPCTL-формулы F на состоянии S произвольной (небазисной) вероятностной МАС A за время, зависящее дважды экспоненциально от |A| и линейно от |F|.

Литература

1. Baier C., Katoen J., Principles of model checking. MIT Press, 2008.
2. Courcoubetis C., Yannakakis M., The complexity of probabilistic verification. J. ACM, v. 42, 4, 1995, 857-907.
3. Dekhtyar A., Dekhtyar M.I. Revisiting the Semantics of Interval Probabilistic Logic Programs. // Proc. 8th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'05), LNAI, V. 3662, 2005. P.330-342.
4. 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.
5. Dekhtyar M.I., Dikovsky A.Ja., Valiev M.K. Temporal Verification of Probabilistic Multi-Agent Systems.// Pillars of Computer Science: Essays Dedicated to Boris (Boaz)Trakhtenbrot on the Occasion of His 85th Birthday, LNCS, N 4800, 2008, 256-265.
6. Dix J., Nanni M., and Subrahmanian V. S. , Probabilistic agent reasoning. ACM Transactions of Computational Logic, 1(2), 2000, 201-245.
7. Hansson H., Jonsson B. A logic for reasning about time and reliability. Formal Aspects of Computing, 6(5), 1994, 512-535.
8. Ng R. and Subrahmanian V.S. Probabilistic Logic Programming. // Information and Computation, 101, 2, 1993. P.150--201.
9. Subrahmanian V. S., Bonatti P., Dix J., et al., Heterogeneous Agent Systems, MIT LPess, 2000.

10. Валиев М.К., Дехтярь М.И., Диковский А.Я, О свойствах многоагентных систем с вероятностными каналами связи.// Труды IV-ой Межд. научно-практической конференции "Интегрированные модели и мягкие вычисления в искусственном интеллекте", М.; Физматлит, Коломна, 2007. С.119--126.
11. Валиев М.К., Дехтярь М.И.. Вероятностные мультиагентные сис - темы: семантика и верификация. Вестник Тверского государственного университета, серия «Прикладная математика»., 35 (95), 2008, 9-22.
12. Кларк Э.М., Грамберг О., Пелед Д.. Верификация моделей программ: Model Checking. МЦНМО, М., 2002.
13. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям. Эдиториал УРСС, М., 2002.