



МУЛЬТИАГЕНТНОЕ ОПИСАНИЕ СЛОЖНОГО ОБЪЕКТА ПО ДОСТОВЕРНОЙ ИНФОРМАЦИИ*

Косовская Т.М.¹

¹ СПбГУ, Санкт-Петербург, Россия

Аннотация

Рассматриваются три задачи составления описания объекта по достоверной информации, полученной несколькими агентами, причём информация, которой располагает каждый агент, не полна. В первой задаче объект характеризуется своими глобальными бинарными признаками и описывается набором значений этих признаков. Во второй и третьей задачах объект представлен как множество своих элементов, характеризуется свойствами этих элементов и отношениями между ними и описывается множеством постоянных атомарных формул исчисления предикатов. Во второй задаче предполагается, что все агенты располагают одинаковыми именами частей объекта. В третьей задаче каждый агент не знает подлинных имён частей объекта и даёт им имена произвольно. Приводятся алгоритмы решения рассмотренных задач и доказываются верхние оценки числа шагов их работы. Для второй и третьей задач приведены модельные примеры работы алгоритмов.

Ключевые слова: мультиагентное описание объекта, формула исчисления предикатов, неполная выводимость формулы, вычислительная сложность алгоритма.

Цитирование: Косовская Т.М. Мультиагентное описание сложного объекта по достоверной информации // Компьютерные инструменты в образовании, 2016. № 4. С. 5–18.

1. ВВЕДЕНИЕ

Разработка моделей взаимодействия группы агентов, обладающих различной информацией об одном и том же объекте, является в настоящее время одной из важных задач искусственного интеллекта, решению которой посвящены многие работы исследователей. Различные постановки задач математического моделирования при мультиагентном подходе изложены, например, в [4, 11, 13]. При этом у каждого автора имеется своё представление о языке моделирования и, следовательно, о том, что такое «основные задачи математического моделирования» [13]. Так, например, эти представления варьируются от построения моделей в виде деревьев решений, рассмотренного в [4], или от редукции баз нечётких правил, использующей метод муравьиных колоний [10].

В [3] предлагается мультиагентное построение нейронной сети, причём предполагается, что отдельный агент может иметь лишь частичное представление о задаче и способен решить лишь некоторую ее подзадачу.

*Работа выполнена при поддержке гранта РФФИ 14-08-01276-а.

Прикладным аспектам использования мультиагентного подхода посвящены, например, [14, 15] (социальные аспекты принятия решения), [12] (анализ изображений), [2] (задача о роботах в пространстве), [1] (исследование сети). Большинство моделей опирается на то, что информация, собранная агентами, формализована в виде строки значений (бинарных или многозначных) признаков или описывается пропозициональными (булевыми) формулами.

Однако оценки сложности предложенных подходов имеются разве лишь в [2]. Модели, описанные в [14, 15], трудны для практического применения, поскольку их создатели не ставят перед собой задачу адекватно оценить время работы компьютерных программ, реализующих эти модели. Возникает ощущение, что многие задачи, которые необходимо решить в рамках этих моделей, являются NP-трудными или даже алгоритмически неразрешимыми.

Интересен подход, представленный в [17], основанный на использовании секвенциального исчисления высказываний и различающий ситуацию, в которой для всех агентов некоторой группы Γ справедливы утверждения A и B (то есть $A \& B$ коллективный выбор группы Γ), от того, что если для группы Γ справедливо утверждение A , а для группы Δ справедливо утверждение B , то для группы $\Gamma \cup \Delta$ справедливо утверждение $A \wedge B$ — выбор некоторых коалиций (Γ и Δ).

В другой работе этого же автора [16] использовано исчисление предикатов, но автор не ставит перед собой задачу полного описания объекта, а лишь констатацию того, какая группа экспертов поддерживает справедливость принятия того или иного решения.

Использование языка исчисления предикатов для решения задач искусственного интеллекта было предложено ещё в середине XX века, (например в [5]) и продолжает описываться в теоретических работах (см., например [11]). Именно язык исчисления предикатов позволяет адекватно описывать сложные объекты, характеризующиеся свойствами своих частей и отношениями между ними.

Однако проблема практического применения такого подхода связана с NP-трудностью возникающих при этом задач. Наличие оценок их временной сложности позволяет для каждой конкретной задачи найти такие ограничения на исходные данные, которые существенно уменьшат время работы алгоритма. Поэтому ниже для каждого алгоритма доказываются верхние оценки их временной сложности.

В настоящей работе предлагаются модели мультиагентного описания объекта как на языке исчисления высказываний (что соответствует стандартным постановкам задачи и не вызывает каких-либо трудностей), так и на языке исчисления предикатов. При этом рассматриваются три типа постановки задачи.

В первой задаче формализация производится на языке исчисления высказываний (булевых формул), а во второй и третьей использован язык исчисления предикатов. В двух последних задачах каждый агент обладает частью описания объекта, но различие в постановке заключается в том, известны ли агенту настоящие (или, по крайней мере, общие для всех агентов) имена частей описываемого объекта или каждый агент вправе называть каждую часть объекта на свой вкус.

Если решение первых двух задач достаточно просто (они приводятся, главным образом, чтобы подчеркнуть трудоёмкость третьей задачи), то третья задача напоминает притчу о трёх слепцах, ощупывающих слона. В постановке третьей задачи предполагается, что каждая пара агентов обладает информацией о некоторой общей части объекта, называя эту часть по-разному. Главная проблема — найти и отождествить эти общие части.

2. ПОСТАНОВКА ЗАДАЧИ МУЛЬТИАГЕНТНОГО ОПИСАНИЯ ОБЪЕКТА ПРИ ЕГО ОПИСАНИИ ГЛОБАЛЬНЫМИ ПРИЗНАКАМИ

Пусть объект ω глобально характеризуется бинарными признаками $p_1, \dots, p_n$. Информация об объекте — это элементарная конъюнкция с переменными $p_1, \dots, p_n$, в которой атом p_i означает, что объект обладает i -ым признаком, атом $\neg p_i$ означает, что объект не обладает i -ым признаком, а отсутствие в ней переменной p_i означает, что значение i -го признака в этой информации не известно.

Имеется m агентов $a_1, \dots, a_m$, каждый из которых обладает информацией $I_1, \dots, I_m$ соответственно. Информация I_j имеет вид $I_j = p_1^{\alpha_1^j} \& \dots \& p_n^{\alpha_n^j}$, где $(\alpha_1^j, \dots, \alpha_n^j)$ — набор значений признаков, причём $\alpha_i^j \in \{0, 1, *\}$ и $*$ означает, что значение i -го признака агентом a_j не измерено. Здесь выражение вида p^α используется как обозначение для p , если $\alpha = 1$, для $\neg p$, если $\alpha = 0$ или для пустого слова, если $\alpha = *$.

Задача 1. Информация, которой обладает каждый агент, абсолютно достоверна. Требуется построить описание объекта в виде строки $(\alpha_1, \dots, \alpha_n)$, где α_i — значение i -го признака.

Решение задачи 1. Пусть j -ый агент ($j = 1, \dots, m$) обладает информацией $I_j = p_1^{\alpha_1^j} \& \dots \& p_n^{\alpha_n^j}$, где $(\alpha_1^j, \dots, \alpha_n^j)$ — набор значений признаков, измеренных этим агентом.

Так как информация, которой обладает каждый агент, предполагается абсолютно достоверной, то разные агенты не могут получить различные значения из $\{0, 1\}$ для одного и того же признака. Поэтому строки значений $(\alpha_1^j, \dots, \alpha_n^j)$ для различных j не могут содержать в одной и той же позиции как 0, так и 1.

Результирующей строкой $(\alpha_1, \dots, \alpha_n)$ значений для признаков будет строка, в которой значение $*$ находится в тех и только тех позициях, в которых все строки содержат $*$. В остальных позициях значение строки $(\alpha_1, \dots, \alpha_n)$ совпадает с отличным от $*$ значением любой из строк $(\alpha_1^j, \dots, \alpha_n^j)$.

Если дополнительно предполагается, что имеется база всех известных описаний объектов в терминах признаков $p_1, \dots, p_n$, то можно дополнить неизвестные значения признаков путём выбора ближайшего соседа с теми же известными значениями признаков из базы в соответствии с некоторой метрикой в пространстве признаков. Например, в соответствии с хорошо известными метриками вида $\|\bar{x} - \bar{y}\| = (\sum_{i=1}^n |x_i - y_i|^k)^{\frac{1}{k}}$.

3. ПОСТАНОВКИ ЗАДАЧ МУЛЬТИАГЕНТНОГО ОПИСАНИЯ СЛОЖНОГО ОБЪЕКТА ПРИ ИСПОЛЬЗОВАНИИ ЛОКАЛЬНЫХ ПРИЗНАКОВ

Пусть объект ω представляет собой множество составляющих его элементов $\omega = \{\omega_1, \dots, \omega_r\}$ и локально характеризуется предикатами $p_1, \dots, p_n$, задающими свойства этих элементов и отношения между ними.

Информация об объекте — это элементарная конъюнкция постоянных литералов с предикатами $p_1, \dots, p_n$.

Имеется m агентов $a_1, \dots, a_m$, которые могут измерить некоторые значения признаков на некоторых элементах объекта ω (то есть определить свойства некоторых элементов исследуемого объекта и некоторые отношения между этими элементами). Каждый

из агентов $a_1, \dots, a_m$ обладает информацией $I_1, \dots, I_m$ соответственно. Информация, которой обладает каждый агент, абсолютно достоверна.

Имеется база данных с полными описаниями всех ранее известных объектов.

Задача 2. Требуется построить описание объекта ω в виде конъюнкции атомарных формул или их отрицаний, задающих свойства элементов заданного объекта и отношения между этими элементами.

Решение задачи 2. Так как информация, которой обладает каждый агент, предполагается абсолютно достоверной, то в элементарной конъюнкции $I_1 \& \dots \& I_m$ нет контрарных членов, но некоторые подформулы могут повторяться.

После удаления повторяющихся подформул получим элементарную конъюнкцию I , содержащую всю информацию, собранную агентами. Если информация полна, то осталось найти в базе данных объект с таким же (с точностью до имён констант) описанием.

Если же агентам не удалось собрать всю информацию об объекте, то в базе данных нужно найти «ближайшего соседа», описание которого не противоречит полученной информации. Алгоритм нахождения такого «ближайшего соседа» имеется, например, в [8].

Задача 3. Требуется построить описание объекта ω в виде конъюнкции атомарных формул или их отрицаний, задающих свойства элементов заданного объекта и отношения между этими элементами, при условии, что агент a_j может не знать реального количества элементов в объекте ω и предполагать, что он имеет дело с объектом $\omega^j = \{\omega_1^j, \dots, \omega_{r_j}^j\}$ (имена элементов у каждого агента свои, например, как если бы агенты рассматривали один и тот же объект с разных сторон и в качестве имён элементов давали им номера в соответствии с порядком исследования своей части объекта).

4. МАКСИМАЛЬНАЯ ОБЩАЯ С ТОЧНОСТЬЮ ДО ИМЁН АРГУМЕНТОВ ПОДФОРМУЛА ДВУХ ЭЛЕМЕНТАРНЫХ КОНЪЮНКЦИЙ

В силу того, что каждый агент использует собственные обозначения для имён элементов объекта, возможность непосредственного объединения всех информации отсутствует. Требуется найти все возможные общие с точностью до имён аргументов подформулы различных информации и унификаторы для них, то есть такие замены имён элементов объекта, что выделенные подформулы станут графически равными.

Определение. Формула C называется максимальной общей с точностью до имён аргументов подформулой двух элементарных конъюнкций A и B , если существуют такие подстановки λ_{CA} и λ_{CB} аргументов формул A и B соответственно вместо аргументов формулы C , что результаты этих подстановок $[C]\lambda_{CA}$ и $[C]\lambda_{CB}$ совпадают с некоторыми подформулами формул A и B соответственно, причём ни для какой элементарной конъюнкции C' с бóльшим количеством литералов такие подстановки не существуют.

Подстановки λ_{CA} и λ_{CB} будем называть общими унификаторами формулы C с формулами A и B соответственно.

Например, две формулы $A = p(a, b) \& p(b, a) \& q(b, a, c)$ и $B = p(b, a) \& p(b, d) \& q(a, b, d)$ имеют общую с точностью до имён аргументов подформулу. Пусть $\{x_1/u_1, x_2/u_2, \dots, x_k/u_k\}$ обозначает замену аргументов $x_1, x_2, \dots, x_k$ на $u_1, u_2, \dots, u_k$ соответственно. При замене $\{a/u, b/v, c/w\}$ в формуле A получим формулу $C_1 = p(u, v) \& p(v, u) \& q(v, u, w)$. Если к формуле B применить замену $\{a/v, b/u, d/w\}$, то получим формулу $C_2 = p(u, v) \& p(u, w) \& q(v, u, w)$.

Общей подформулой формул C_1 и C_2 будет формула $C = p(u, v) \& q(v, u, w)$, которая с точностью до имён переменных является общей подформулой формул A и B с унификаторами соответственно $\lambda_{CA} = \{u/a, v/b, w/c\}$ и $\lambda_{CB} = \{u/b, v/a, w/d\}$.

В работе [7] предложено понятие неполной выводимости формулы B из заданного множества формул S . Оно заключается в том, что вместо проверки логического следования $S \Rightarrow \exists x_{\neq} B'$, (где S — набор постоянных литералов, B — конъюнкция литералов, не содержащих константы), ищется такая подформула B' формулы B и такой подписок x' списка всех переменных формулы B , что имеет место $S \Rightarrow \exists x'_{\neq} B'^1$. При этом для обозначения того, что из списка S частично следует формула B , используется выражение $S \Rightarrow_P B$.

Для нахождения B' последовательно удаляем литералы из формулы B и проверяем, имеется ли требуемое логическое следование. Такая проверка возможна как с помощью переборного алгоритма (перебираются все возможные подстановки констант из S вместо переменных из B' и проверяется, что каждый полученный литерал содержится в S), так и с помощью логических методов доказательства, например, построения вывода в секвенциальном исчислении предикатов или доказательства методом резолюций для исчисления предикатов. Все эти методы доказательства конструктивны в том смысле, что в случае нахождения требуемой подформулы B' не только доказано, что $S \Rightarrow \exists x'_{\neq} B'$, но и предъявляется такой набор констант из S , после подстановки которого вместо переменных формулы B' , каждый полученный литерал содержится в S .

Такую подстановку будем называть максимальным общим унификатором формулы B и списка литералов S .

Понятие неполной выводимости формулы из списка постоянных литералов можно применить к нахождению максимальной общей с точностью до имён аргументов подформулы двух элементарных конъюнкций A и B . Для этого достаточно все константы в формуле B заменить на переменные (разные константы заменяем на разные переменные и получаем B^*), в элементарной конъюнкции A удалить знак $\&$ между литералами (получаем A^*) и проверить неполное следование $A^* \Rightarrow_P B^*$.

Следует отметить, что при проверке $A^* \Rightarrow_P B^*$ и $B^* \Rightarrow_P A^*$ выделяется одна и та же с точностью до имён переменных подформула. Поэтому не будем различать эти два частичные логические следования, а при подсчёте числа шагов проверки будем считать, что в правой части находится формула с меньшим числом аргументов при использовании алгоритма полного перебора или с меньшим числом литералов при использовании логических методов проверки логического следования (эти параметры находятся в показателе степени оценки числа шагов соответствующих алгоритмов [6]).

Решение задачи 3. Опишем алгоритм получения полной информации об объекте по имеющимся у агентов $a_1, \dots, a_m$ информациям $I_1, \dots, I_m$ соответственно.

1. Заменяем все константы в $I_1, \dots, I_m$ на переменные так, чтобы разные константы были заменены на разные переменные и имена переменных в списках переменных для I_i и I_j при $i \neq j$ не совпадали. Получаем список элементарных конъюнкций $I'_1, \dots, I'_m$.
2. Для каждой пары формул I'_i и I'_j ($i = 1, \dots, m-1, j = i+1, \dots, m$) проверяем неполную выводимость $I_i \Rightarrow_P I_j$ с выделением их максимальной общей с точностью до имён аргументов подформулы C_{ij} и унификаторов $\lambda_{i,ij}$ и $\lambda_{j,ij}$. Всем аргументам C_{ij} даём уникальные имена.

¹Выражение $\exists x_{\neq} B'$ используется для обозначения формулы $\exists x(\&_{i=1}^{m-1} \&_{j=i+1}^m x_i \neq x_j \& B)$, где x_i и x_j — переменные из списка переменных δ , и читается как «существуют различные значения для переменных из списка x ».

3. Дополнительно для каждой пары $i, j (i > j)$ проверяем, не получились ли в I'_i и I'_j после применения к ним соответственно унификаторов $\lambda_{i,ij}$ и $\lambda_{j,ij}$ противоречивые литералы. То есть либо контрарная пара литералов, либо противоречивая пара литералов в соответствии с определением исходных предикатов. Если такое противоречие выявлено, то удаляем из C_{ij} литералы, в которых присутствуют переменные, подстановка которых привела к появлению противоречивых литералов. Изменяем унификаторы, исключив из них эти переменные вместе с содержащими эти переменные литералами, и применяем к исходным формулам I'_i и I'_j новые унификаторы.
4. Для каждого i отождествляем переменные в $C_{ij} (i \neq j)$, которые подставляются в I'_i и I'_j вместо одной и той же исходной переменной. Заменяем в подстановочной части унификаторов имена отождествлённых переменных на одно и то же имя.
5. Применяем полученные в п. 4 унификаторы к $I'_1, \dots, I'_m$. Получаем $I''_1, \dots, I''_m$.
6. Составляем $I''_1 \& \dots \& I''_m$ и удаляем повторяющиеся конъюнктивные члены. Получаем описание I .
7. Проверяем наличие полученного описания в базе. При этом если описание I содержит исходные переменные, введённые в п. 1, то, возможно, потребуется их отождествление либо между собой, либо с вновь полученными переменными.

Корректность алгоритма решения задачи 3. Предложенный алгоритм выделяет максимально возможное описание исследуемого объекта.

П. 1 алгоритма необходим для возможности применения проверки неполной выводимости, а также для избежания ситуации, когда разные агенты дали одинаковые имена разным элементам объекта.

В п. 2 алгоритма попарно выделяются фрагменты объекта, описания которых совпадают у пар агентов. При таком выделении может оказаться, что в этих описаниях отсутствуют те подформулы, которые осуществляют «связь» между переменными из выделенных фрагментов и остальными переменными в информации каждого из агентов этой пары. Эти подформулы после применения к ним соответствующих унификаторов могут привести к противоречивости информации. Для избежания этого используется п. 3 алгоритма.

В п.п. 4 и 5 попарно отождествляются имена переменных, составляющих непротиворечивые фрагменты, имеющие одинаковые описания у различных агентов. Тем самым получаем описания, аналогичные таковым в постановке задачи 1.

При этом может возникнуть ситуация (см., например, рис. 5 в приведённом ниже примере и соответствующее ему описание объекта), когда некоторые переменные из описаний $I'_1, \dots, I'_m$ не вошли в полученные унификаторы и, следовательно, сохранились в $I''_1, \dots, I''_m$. Это означает, что информация, собранная агентами, не позволяет отождествить эти переменные ни между собой, ни с какими-либо другими переменными. В п. 7 алгоритма предлагается использовать какие-либо эвристические методы для отождествления этих переменных либо между собой, либо с другими переменными, основываясь на тех описаниях, которые имеются в базе.

5. ОЦЕНКИ ЧИСЛА ШАГОВ РЕШЕНИЯ ПОСТАВЛЕННЫХ ЗАДАЧ

Решение задачи 1 сводится к просмотру m строк, состоящих из n символов, входящих в множество $\{0, 1, *\}$ и записи в результирующую строку символа 0 или 1 при первой встрече одного из них в столбце. Если же все символы в столбце совпадают с $*$, то в ре-

зультатирующую строку записывается *. Очевидно, что число «шагов» этой процедуры не превосходит $n(m+1)$. Здесь под шагом понимается проверка значения символа или запись символа в результирующую строку.

Если дополнительно требуется найти объект в базе данных, отличающийся от полученной строки только в тех позициях, в которых получено значение *, то добавляется время, необходимое на поиск в базе, и зависящее от размера базы и способа организации данных в ней.

Решение задачи 2 сводится к удалению повторяющихся конъюнктивных членов в элементарной конъюнкции $I_1 \& \dots \& I_m$. Если $\|I_j\|$ — количество атомарных формул в I_j , то число «шагов» этой процедуры не превосходит

$$\sum_{i=1}^{m-1} \sum_{j=i+1}^m \|I_i\| \|I_j\| \leq \frac{m(m-1)}{2} (\max_j \|I_j\|)^2.$$

Здесь под шагом понимается сравнение двух атомарных формул на графическое совпадение.

Общее число шагов решения задачи 2 составляет $O(m^2 \|I\|^2)$, где $\|I\|$ — максимальное количество атомарных формул в $I_1, \dots, I_m$.

Если дополнительно требуется найти описанный объект в базе данных, то число шагов увеличится на число, необходимое на поиск в базе и зависящее от размера базы и способа организации данных в ней.

Для оценки числа шагов решения задачи 3 оценим число шагов выполнения каждого пункта по отдельности. Ниже под «шагом» работы алгоритма понимается любое из следующих действий: подстановка аргументов вместо переменных/констант; проверка двух литералов или двух переменных на их графическое совпадение.

1. Замена констант на переменные требует $\sum_{j=1}^m \|Arg(I_j)\|$ «шагов», где $\|Arg(I_j)\|$ — количество вхождений аргументов в элементарную конъюнкцию I_j .
2. Проверка неполной выводимости $I_i \Rightarrow_P I_j$ требует $O(t_i^{t_j} \cdot 2^{\|I_j\|})$ «шагов» для алгоритма полного перебора, где t_i — число аргументов элементарной конъюнкции I_i , и $O(\|I_i\|^{\|I_j\|} \cdot \|I_i\|^3)$ для алгоритма, основанного на поиске вывода в секвенциальном исчислении предикатов, или методом резолюций для исчисления предикатов. (Эти оценки получены в [6]). Полученные оценки следует суммировать по $i = 1, \dots, m-1, j = i, \dots, m$. Всего получаем $O(t \cdot 2^{\|I\|} m^2)$ «шагов» для алгоритма полного перебора, где t и $\|I\|$ — максимальные количества аргументов и атомарных формул в $I_i (i = 1, \dots, m)$ соответственно. Для алгоритма, основанного на поиске вывода в секвенциальном исчислении предикатов, или методом резолюций для исчисления предикатов эта оценка составит $O(\|I\|^{\|I\|+3} \cdot m^2)$.
3. Проверка на непротиворечивость двух элементарных конъюнкций I'_i и I'_j потребует $\|I'_i\| \|I'_j\|$ «шагов». Всего выполнение этого пункта алгоритма требует не более $\sum_{i=1}^m (m-1) \|I_i\|$ «шагов», что составляет $O(m^2 \|I\|)$ «шагов».
4. Для каждого i отождествление переменных в $C_{ij} (i > j)$ заключается в просмотре подставляемой части унификаторов и проверке на совпадение заменяемой. Это потребует не более $(m-i) t_i^2$ «шагов». Всего выполнение этого пункта алгоритма требует не более $\sum_{i=1}^m (m-1) t_i^2$ «шагов», что составляет $O(m^2 r^2)$ «шагов».
5. Число «шагов» замены имён переменных линейно относительно $\sum_{i=1}^m \|I_i\|$, что составляет $O(m \|I\|)$ «шагов».

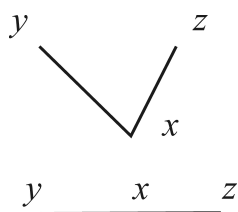
6. Число «шагов» удаления повторяющихся конъюнктивных членов не превосходит $\sum_{i=1}^{m-1} \sum_{j=i+1}^m \|I_i\| \|I_j\|$ и составляет $O(m^2 \|I\|^2)$.
7. Число шагов зависит от числа шагов, необходимых на поиск в базе и зависящих от размера базы и способа организации данных в ней. В случае если описание I содержит исходные переменные, введённые в п. 1, то это число шагов следует умножить на величину, не превосходящую $n_1 n_2$, где n_1 и n_2 — количества исходных переменных и всех переменных в полученном описании соответственно.

Всего число «шагов» работы алгоритма решения задачи 3 составляет $O(t^t m^2 \|I\|^2)$ «шагов» для алгоритма полного перебора, где t и $\|I\|$ — максимальные количества аргументов и атомарных формул в I_j ($j = 1, \dots, m$) соответственно и $O(\|I\|^{\|I\|+3} m^2)$ для алгоритма, основанного на поиске вывода в секвенциальном исчислении предикатов, или методом резолюций для исчисления предикатов.

Анализ процедуры получения этой оценки показывает, что главный вклад в неё даёт количество произведённых проверок неполной выводимости.

6. ПРИМЕРЫ

Рассмотрим решение задач 2 и 3 на примере описания контурного изображения «ящика» в терминах предикатов V и L , задающих отношения между вершинами, представленные на рис. 1.



$$V(x, y, z) \Leftrightarrow (\angle yxz < \pi)$$

$$L(x, y, z) \Leftrightarrow \text{«}x \text{ лежит на отрезке с концами } y \text{ и } z\text{»}$$

Рис. 1. Исходные предикаты

Очевидно, что эти предикаты характеризуют расположение вершины x относительно y и z , а также обладают следующими свойствами: $V(x, y, z) \Leftrightarrow \neg V(x, z, y) \vee L(x, y, z)$ (и, следовательно, $V(x, y, z) \wedge \neg V(x, z, y)$ является противоречием), $L(x, y, z) \Leftrightarrow L(x, z, y)$.

Три агента исследуют различные фрагменты изображения, представленного на рис. 2.

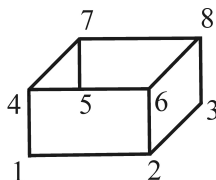


Рис. 2. Исследуемое изображение

Это изображение имеется в базе данных и имеет описание

$$V(1, 4, 2) \wedge V(2, 1, 6) \wedge V(2, 6, 3) \wedge V(2, 1, 3) \wedge V(3, 2, 8) \wedge V(4, 5, 1) \wedge V(4, 6, 1) \wedge V(4, 7, 5) \wedge V(4, 7, 6) \wedge V(4, 7, 1) \wedge V(5, 4, 7) \wedge V(5, 7, 6) \wedge V(6, 2, 5) \wedge V(6, 2, 4) \wedge V(6, 5, 8) \wedge V(6, 4, 8) \wedge V(6, 8, 2) \wedge V(7, 5, 4) \wedge V(7, 8, 5) \wedge V(7, 8, 4) \wedge V(8, 3, 6) \wedge V(8, 6, 7) \wedge V(8, 3, 7) \wedge L(5, 4, 6).$$

Решение задачи 2. Каждый из агентов получил соответственно описания фрагментов изображения, представленных на рис. 3.

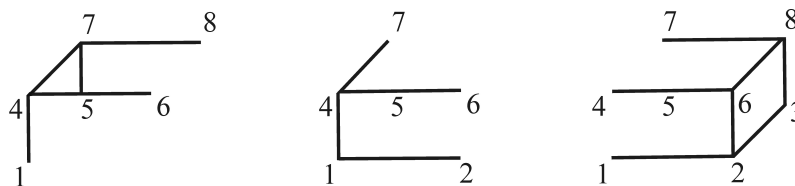


Рис. 3. Фрагменты изображения, полученные агентами при решении задачи 2

Каждое из этих описаний имеет соответственно вид.

$$I_1 = V(4, 5, 1) \& V(4, 6, 1) \& V(4, 7, 5) \& V(4, 7, 6) \& V(4, 7, 1) \& V(5, 4, 7) \& V(5, 7, 6) \& L(5, 4, 6) \& V(7, 5, 4) \& V(7, 8, 5) \& V(7, 8, 4),$$

$$I_2 = V(1, 4, 2) \& V(4, 5, 1) \& V(4, 6, 1) \& V(4, 7, 5) \& V(4, 7, 6) \& V(4, 7, 1) \& L(5, 4, 6),$$

$$I_3 = V(2, 1, 6) \& V(2, 6, 3) \& V(2, 1, 3) \& V(6, 2, 5) \& V(6, 2, 4) \& V(6, 5, 8) \& V(6, 4, 8) \& V(6, 8, 2) \& V(8, 3, 6) \& V(8, 6, 7) \& V(8, 3, 7).$$

Конъюнкция этих описаний даёт описание объекта, имеющегося в базе.

Решение задачи 3.² Каждый из агентов получил соответственно описания фрагментов изображения, представленных на рис. 4.

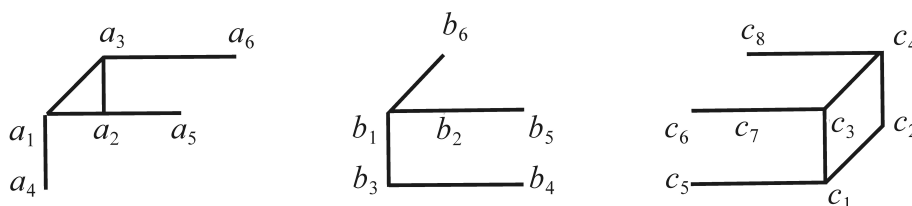


Рис. 4. Фрагменты изображения, полученные агентами при решении задачи 3

Поскольку при решении этой задачи важны имена аргументов у информации, полученной агентом, то будем их явно выписывать.

В соответствии с п. 1 алгоритма, заменяем все константы на переменные, так чтобы разные константы были заменены на разные переменные. При этом каждое из этих описаний имеет соответственно вид:

$$I_1(x_1, \dots, x_6) = V(x_1, x_2, x_4) \& V(x_1, x_5, x_4) \& V(x_1, x_3, x_2) \& V(x_1, x_3, x_5) \& V(x_1, x_3, x_4) \& V(x_2, x_1, x_3) \& V(x_2, x_3, x_5) \& V(x_3, x_2, x_1) \& V(x_3, x_6, x_2) \& V(x_3, x_6, x_1) \& L(x_2, x_1, x_5),$$

$$I_2(y_1, \dots, y_6) = V(y_3, y_1, y_4) \& V(y_1, y_2, y_3) \& V(y_1, y_5, y_3) \& V(y_1, y_6, y_2) \& V(y_1, y_6, y_5) \& V(y_1, y_6, y_3) \& L(y_2, y_1, y_5),$$

$$I_3(z_1, \dots, z_8) = V(z_1, z_5, z_3) \& V(z_1, z_3, z_2) \& V(z_1, z_5, z_2) \& V(z_3, z_1, z_7) \& V(z_3, z_1, z_6) \& V(z_3, z_7, z_4) \& V(z_3, z_6, z_4) \& V(z_3, z_4, z_1) \& V(z_4, z_2, z_3) \& V(z_4, z_3, z_8) \& V(z_4, z_2, z_8) \& L(z_7, z_6, z_3).$$

Выполняем п. 2 алгоритма. Попарная проверка неполной выводимости для этих описаний даёт их максимальные общие с точностью до имён переменных подформулы.

²В этом примере выделение максимальной общей с точностью до имён переменных подформулы двух формул, нахождение общего унификатора и выражение формул через выделенные подформулы производилось с помощью программы, написанной студентом математико-механического факультета Санкт-Петербургского государственного университета Петровым Д.А.

Максимальная с точностью до имён переменных подформула формул $I_1(x_1, \dots, x_6)$ и $I_2(y_1, \dots, y_6)$

$$C_{12}(u_0, \dots, u_4) = V(u_0, u_1, u_2) \& V(u_0, u_3, u_2) \& V(u_0, u_4, u_1) \& V(u_0, u_4, u_3) \& V(u_0, u_4, u_2) \& \\ L(u_1, u_0, u_3)$$

имеет унификаторы $\lambda I_1, C_{12} = \{x_1/u_0, x_2/u_1, x_3/u_4, x_4/u_2, x_5/u_3\}$ и

$\lambda I_2, C_{12} = \{y_1/u_0, y_2/u_1, y_3/u_2, y_5/u_3, y_6/u_3\}$. При этом

$$I_1(u_0, u_1, u_2, u_3, u_4, x_6) = V(u_1, u_0, u_4) \& V(u_1, u_4, u_3) \& V(u_4, u_1, u_0) \& V(u_4, x_6, u_1) \& V(u_4, x_6, u_0) \& \\ C_{12}(u_0, \dots, u_4),$$

$$I_2(u_0, u_1, u_2, y_4, u_3, u_4) = V(u_2, u_0, y_4) \& C_{12}(u_0, \dots, u_4).$$

Максимальная с точностью до имён переменных подформула формул $I_2(y_1, \dots, y_6)$ и $I_3(z_1, \dots, z_8)$

$$C_{23}(v_0, v_2, v_4, v_5, v_6, v_7) = V(v_6, v_2, v_7) \& V(v_2, v_4, v_6) \& V(v_2, v_5, v_6) \& V(v_2, v_0, v_4) \& V(v_2, v_0, v_5) \& \\ \& L(v_4, v_2, v_5)$$

имеет унификаторы $I_2, C_{23} = \{y_1/v_2, y_2/v_4, y_3/v_6, y_4/v_7, y_5/v_5, y_6/v_0\}$ и

$I_3, C_{23} = \{z_1/v_0, z_3/v_2, z_5/v_6, z_6/v_5, z_7/v_4, z_8/v_7\}$. При этом

$$I_2(v_2, v_4, v_6, v_7, v_5, v_0) = V(v_2, v_0, v_6) \& L(v_4, v_2, v_5) \& C_{23}(v_0, v_2, v_4, v_5, v_6, v_7)$$

$$I_3(v_0, z_2, v_2, v_6, z_5, v_5, v_4, v_7) = V(v_2, v_6, v_0) \& V(v_0, z_5, v_2) \& V(v_0, v_2, z_2) \& V(v_0, v_5, z_2) \& \\ V(v_6, z_2, v_2) \& V(v_6, v_2, v_7) \& L(v, v_5, v_2) \& C_{23}(v_0, v_2, v_4, v_5, v_6, v_7).$$

Так как $I_2(v_2, v_4, v_6, v_7, v_5, v_0)$ содержит $V(v_2, v_0, v_6)$, а $I_3(v_0, z_2, v_2, v_6, z_5, v_5, v_4, v_7)$ содержит $V(v_2, v_6, v_0)$, причём в силу определения предиката V формула $V(x, y, z) \& V(x, z, y)$ противоречива, то подстановки, соответствующие унификаторам с этой выделенной подформулой, не могут дать непротиворечивого описания объекта.

После удаления переменных y_1 и z_3 (вместо которых подставлялась новая переменная v_2 , приведшая к противоречию) из I_2 и I_3 соответственно получаем новую их максимальную подформулу

$$C'_{23}(v_0, v_1, v_2) = L(v_1, v_0, v_2)$$

и унификаторы $\lambda_{I_2, C'_{23}} = \{y_1/v_0, y_2/v_1, y_3/v_2\}$ и $\lambda_{I_3, C'_{23}} = \{z_3/v_2, z_6/v_2, z_7/v_2\}$. При этом

$$I_2(v_0, v_1, v_2, y_4, y_5, y_6) = V(v_2, v_0, y_4) \& V(v_0, v_1, v_2) \& V(v_0, y_5, v_2) \& V(v_0, y_6, v_1) \& V(v_0, y_6, y_5) \& \\ V(v_0, y_6, v_2) \& C'_{23}(v_0, v_1, v_2),$$

$$I_3(z_1, z_2, v_2, z_4, z_5, v_0, v_1, z_8) = V(z_1, z_5, v_2) \& V(z_1, v_2, z_2) \& V(z_1, z_5, z_2) \& V(v_2, z_1, v_1) \& V(v_2, z_1, v_0) \& \\ \& V(v_2, v_1, z_4) \& V(v_2, v_0, z_4) \& V(v_2, z_4, z_1) \& V(z_4, z_2, v_2) \& \\ \& V(z_4, v_2, z_8) \& V(z_4, z_2, z_8) \& C'_{23}(v_0, v_1, v_2).$$

Максимальная с точностью до имён переменных подформула формул $I_1(x_1, \dots, x_6)$ и $I_3(z_1, \dots, z_8)$

$$C_{13}(w_0, \dots, w_6) = V(w_2, w_4, w_6) \& V(w_2, w_5, w_6) \& V(w_2, w_0, w_4) \& V(w_2, w_0, w_5) \& V(w_0, w_1, w_2)$$

имеет унификаторы $\lambda_{I_1, C_{13}} = \{x_1/w_2, x_2/w_4, x_3/w_0, x_4/w_6, x_5/w_5, x_6/w_1\}$

и $\lambda_{I_3, C_{13}} = \{z_1/w_0, z_3/w_2, z_4/w_6, z_5/w_1, z_6/w_5, z_7/w_2\}$. При этом

$$I_1(w_2, w_4, w_0, w_6, w_5, w_1) = V(w_2, w_0, w_6) \& V(w_0, w_1, w_4) \& V(w_0, w_4, w_2) \& L(w_2, w_4, w_5) \& \\ C_{13}(w_0, \dots, w_6),$$

$$I_3(w_0, z_2, w_2, w_6, w_1, w_5, w_4, z_8) = V(w_0, w_2, w_3) \& V(w_0, w_1, w_3) \& V(w_2, w_6, w_0) \& V(w_6, w_3, w_2) \& \\ \& V(w_6, w_2, w_7) \& V(w_6, w_3, w_7) \& C_{13}(w_0, \dots, w_6).$$

Так как $I_1(w_2, w_4, w_0, w_6, w_5, w_1)$ содержит $V(w_2, w_0, w_6)$, а $I_3(w_0, z_2, w_2, w_6, w_1, w_5, w_4, z_8)$ содержит $V(w_2, w_6, w_0)$, причём в силу определения предиката V формула

$V(x, y, z) \& V(x, z, y)$ противоречива, то подстановки, соответствующие унификаторам с этой выделенной подформулой, не могут дать непротиворечивого описания объекта.

После удаления переменных x_1 и z_3 (вместо которых подставлялась новая переменная w_2 , приведшая к противоречию) из I_1 и I_3 соответственно получаем новую их максимальную подформулу

$$C'_{13}(w_0, w_1, w_2) = L(w_1, w_0, w_2)$$

и унификаторы $\lambda_{I_1, C'_{13}} = \{x_1/w_0, x_2/w_1, x_5/w_2\}$ и $\lambda_{I_3, C'_{13}} = \{z_3/w_2, z_4/w_1, z_5/w_2\}$ — подстановка w_2, w_1, w_0 вместо z_3, z_4, z_5 соответственно. При этом

$$\begin{aligned} I_1(w_0, w_1, x_3, x_4, w_2, x_6) &= V(w_0, w_1, x_4) \& V(w_0, w_2, x_4) \& V(w_0, x_3, w_1) \& V(w_0, x_3, w_2) \& \\ &V(w_0, x_3, x_4) \& V(w_1, w_0, x_3) \& V(w_1, x_3, w_2) \& V(x_3, w_1, w_0) \& \\ &V(x_3, x_6, w_1) \& V(x_3, x_6, w_0) \& C'_{13}(w_0, w_1, w_2), \\ I_3(z_1, z_1, w_2, w_1, w_0, z_6, z_7, z_8) &= V(z_1, w_0, w_2) \& V(z_1, w_2, z_2) \& V(z_1, w_0, z_2) \& V(w_2, z_1, z_7) \& \\ &V(w_2, z_1, z_6) \& V(w_2, z_7, w_1) \& V(w_2, z_6, w_1) \& V(w_2, w_1, z_1) \& \\ &V(w_1, z_2, w_2) \& V(w_1, w_2, z_8) \& V(w_1, z_2, z_8) \& C'_{13}(w_0, w_1, w_2). \end{aligned}$$

В соответствии с п. 4 алгоритма в полученных унификаторах $\lambda_{I_1, C'_{13}}, \lambda_{I_2, C'_{12}}, \lambda_{I_2, C'_{23}}, \lambda_{I_3, C'_{23}}, \lambda_{I_1, C'_{13}}$ и I_3, C'_{13} отождествляем новые переменные, подставляемые вместо одной и той же исходной переменной. То есть отождествляем переменные

u_0 и w_0 (подставляются вместо переменной x_1),
 u_1 и w_1 (подставляются вместо переменной x_2),
 u_2 и w_2 (подставляются вместо переменной x_4),
 u_0 и v_0 (подставляются вместо переменной y_1),
 u_1 и v_1 (подставляются вместо переменной y_2),
 u_2 и v_2 (подставляются вместо переменной y_3),
 v_0 и w_0 (подставляются вместо переменной z_6),
 v_1 и w_1 (подставляются вместо переменной z_3),
 v_2 и w_2 (подставляются вместо переменной z_7).

При этом вместо переменных u_0, v_0 и w_0 пишем 0, вместо переменных u_1, v_1 и w_1 пишем 1, вместо переменных u_2, v_2 и w_2 пишем 2. В результате получили следующие описания

$$\begin{aligned} I_1(\alpha_0, \alpha_1, u_4, u_2, \alpha_2, x_6) &= V(\alpha_0, \alpha_1, u_2) \& V(\alpha_0, \alpha_2, u_2) \& V(\alpha_0, u_4, \alpha_1) \& V(\alpha_0, u_4, \alpha_2) \& V(\alpha_0, u_4, u_2) \\ &\& V(\alpha_1, \alpha_0, u_4) \& V(\alpha_1, u_4, \alpha_2) \& L(\alpha_1, \alpha_0, \alpha_2) \& V(x_3, \alpha_1, \alpha_0) \& \\ &V(u_4, x_6, \alpha_1) \& V(u_4, x_6, \alpha_0), \\ I_2(\alpha_0, \alpha_1, u_2, y_4, \alpha_2, u_4) &= V(u_2, \alpha_0, y_4) \& V(\alpha_0, \alpha_1, u_2) \& V(\alpha_0, \alpha_2, u_2) \& V(\alpha_0, u_4, \alpha_1) \& V(\alpha_0, u_4, \alpha_2) \\ &\& V(\alpha_0, u_4, u_2) \& L(\alpha_1, \alpha_0, \alpha_2), \\ I_3(z_1, z_2, \alpha_2, z_4, z_5, \alpha_0, \alpha_1, z_8) &= V(z_1, z_5, \alpha_2) \& V(z_1, \alpha_2, z_2) \& V(z_1, z_5, z_2) \& V(\alpha_2, z_1, \alpha_1) \& \\ &V(\alpha_2, z_1, \alpha_0) \& V(\alpha_2, \alpha_1, z_4) \& V(\alpha_2, \alpha_0, z_4) \& V(\alpha_2, z_4, z_1) \& \\ &V(z_4, z_2, \alpha_2) \& V(z_4, \alpha_2, z_8) \& V(z_4, z_2, z_8) \& L(\alpha_1, \alpha_0, \alpha_2). \end{aligned}$$

Их конъюнкция

$$\begin{aligned} &V(\alpha_0, \alpha_1, u_2) \& V(\alpha_0, \alpha_2, u_2) \& V(\alpha_0, u_4, \alpha_1) \& V(\alpha_0, u_4, \alpha_2) \& V(\alpha_0, u_4, u_2) \& V(\alpha_1, \alpha_0, u_4) \& \\ &V(\alpha_1, u_4, \alpha_2) \& V(x_3, \alpha_1, \alpha_0) \& V(u_4, x_6, \alpha_1) \& V(u_4, x_6, \alpha_0) \& V(u_2, \alpha_0, y_4) \& V(z_1, z_5, \alpha_2) \& \\ &V(z_1, \alpha_2, z_2) \& V(z_1, z_5, z_2) \& V(\alpha_2, z_1, \alpha_1) \& V(\alpha_2, z_1, \alpha_0) \& V(\alpha_2, \alpha_1, z_4) \& V(\alpha_2, \alpha_0, z_4) \& \\ &V(\alpha_2, z_4, z_1) \& V(z_4, z_2, \alpha_2) \& V(z_4, \alpha_2, z_8) \& V(z_4, z_2, z_8) \& L(\alpha_1, \alpha_0, \alpha_2) \end{aligned}$$

позволяет «склеить» эти изображения по одинаковым переменным. Изображение, соответствующее результату такой «склейки», представлено на рис. 5.

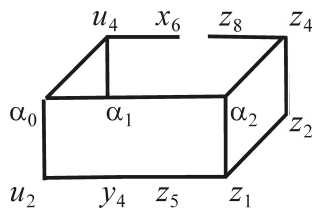


Рис. 5. Изображение, соответствующее результату «склейки»

Осталось отождествить некоторые из исходных переменных ($x_6, y_4, z_1, z_2, z_4, z_5, z_8$) с другими переменными, чтобы получить описание объекта, имеющегося в базе. Такое отождествление можно осуществить перебором. Описание объекта из базы получится при отождествлении следующих пар переменных: $x_6 = z_4, y_4 = z_1, z_5 = u_2, z_8 = u_4$.

При этом переменным присвоены следующие значения: $x_6 = z_4 = 8, y_4 = z_1 = 2, z_5 = u_2 = 1, z_8 = u_4 = 8, z_2 = 3, \alpha_0 = 4, \alpha_1 = 5, \alpha_2 = 6$.

7. ЗАКЛЮЧЕНИЕ

В работе, в частности, предложен алгоритм решения достаточно сложной задачи мультиагентного описания на языке исчисления предикатов сложного составного объекта при условии, что разные агенты могут по-разному называть одни и те же элементы объекта. Описания таких объектов на языке исчисления высказываний достаточно громоздки, и длина их записи экспоненциальна по сравнению с длиной записи на языке исчисления предикатов [11]. Это объясняет естественность того, что решение этой задачи экспоненциально по сравнению с доказанными в работе оценками числа шагов мультиагентного описания объекта на языке исчисления высказываний.

Анализ полученных в работе оценок сложности предложенных алгоритмов позволяет накладывать условия на исходные предикаты с целью снижения реального времени работы алгоритма. Так, например, наличие большого количества исходных предикатов-признаков, каждый из которых достаточно редко встречается в описаниях, обеспечивает уменьшение числа шагов алгоритма.

В дальнейшем предполагается применить предложенные алгоритмы к задачам мультиагентного описания объекта при условии, что информация, предоставленная агентами, может быть не достоверной.

Список литературы

1. Березин А.В. Многоагентная система мониторинга процессов корпоративной сети // Системы управления и информационные технологии, 2014. Т. 55. № 1. С. 47–51.
2. Бернштейн А.Ю., Шилов Н.В. Мультиагентная задача о роботах в пространстве: сложностной, информационный и криптографический аспекты // Моделирование и анализ информационных систем, 2013. Т. 20. № 2. С. 34–53.
3. Болгов А.Н. Мультиагентный подход в реконструкции текстуры на изображениях // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева, 2014. № 5 (57). С. 44–50.
4. Гофман Е.А., Олейник А.А., Субботин С.А. Агентный метод синтеза деревьев решений // Вестник Национального технического университета Харьковский политехнический институт. Серия: Информатика и моделирование, 2011. № 17. С. 16–25.
5. Дуда Р., Харт П. Распознавание образов и анализ сцен. М.: Мир, 1976.

6. Косовская Т.М. Некоторые задачи искусственного интеллекта, допускающие формализацию на языке исчисления предикатов, и оценки числа шагов их решения // Труды СПИИРАН, 2010. Вып. 14. С. 58–75.
7. Косовская Т.М. Частичная выводимость предикатных формул как средство распознавания объектов с неполной информацией // Вестн. С.-Петербург. ун-та. Сер. 10. 2009. Вып. 1. С. 74–84.
8. Kosovskaya T. Distance between objects described by predicate formulas // International Book Series. Information Science and Computing. Book 25. Mathematics of Distances and Applications (Michel Deza, Michel Petitjean, Krasimir Markov (eds)), ITHEA — Publisher, Sofia, Bulgaria, 2012. P. 153–159.
9. Косовская Т.М. Подход к решению задачи построения многоуровневого описания классов на языке исчисления предикатов // Труды СПИИРАН, 2014. № 3 (34). С. 204–217.
10. Олейник А.А., Субботин С.А. Редукция баз нечётких правил на основе мультиагентного подхода // Вестник Национального технического университета Харьковский политехнический институт. Серия: Информатика и моделирование, 2009. № 43. С. 126–137.
11. Рассел С., Норвиг П. Искусственный интеллект: современный подход. 2-е изд.: Пер. с англ. М.: Издательский дом «Вильямс», 2006.
12. Цибульский Г.М. Мультиагентный подход к анализу изображений. Новосибирск: Краснояр. гос. техн. ун-т, 2005.
13. Черемисина Е.В. Постановка основных задач математического моделирования при мультиагентном подходе // Вестник Тамбовского университета. Серия: Естественные и технические науки, 2012. Т. 17. № 1. С. 174.
14. Artikis A., Sergot M., Paliouras G. A logic-based approach to activity recognition // In Hans W. Guesgen and Stephen Marsland, editors, Human Behavior Recognition Technologies: Intelligent Applications for Monitoring and Security, Chapter 1, p. 1–13. Information Science Reference, 2013.
15. Lorini E., Longin D., Mayor E. A logical analysis of responsibility attribution: emotions, individuals and collectives. // Journal of Logic and Computation, 2014. № 24(6). P. 1313–1339.
16. Porello D., Endriss U. Ontology Merging as Social Choice: Judgment Aggregation under the Open World Assumption // Journal of Logic and Computation, 2014. № 24 (6). P. 1229–1249.
17. Porello D. Logics for Collective Reasoning // Proc. of the European Conference on Social Intelligence (ECSI 2014). November. P. 148–159.

Поступила в редакцию 10.03.2016, окончательный вариант — 28.07.2016.

Computer tools in education, 2016

№ 4: 5–18

<http://ipo.spb.ru/journal>

MULTI-AGENT DESCRIPTION OF A COMPLEX OBJECT BASED ON CERTAIN INFORMATION

Kosovskaya T.M.¹

¹SPbSU, Saint-Petersburg, Russia

Abstract

Three problems of an object description based on certain incomplete information received by several agents are under consideration. An object in the first problem is characterized by global binary features and is described by a string of these features values. An object in the second and in the third problems is presented as a set of its elements and is characterized by properties of these elements and relations between them. It is described by a set of constant atomic predicate formulas. It is supposed that all agents in the

second problem have the same name for every object element. In the third problem every agent does not know the true names of the object elements and arbitrarily gives names for them. Algorithms solving the set problems are described and the upper bounds of these algorithms run steps are proved. Model examples of an algorithm implementation are given for the second and the third problems.

Keywords: *multi-agent description of an object, predicate formula, partial deduction, computational complexity of an algorithm.*

Citation: Kosovskaya, T., 2016. "Mul'tiagentnoe opisanie slozhnogo ob"ekta po dostovernoi informatsii" ["Multi-agent Description of a Complex Object Based on Certain Information"], Computer tools in education, no. 4. pp. 5–18.

Received 10.03.2016, the final version — 28.07.2016.

**Tatiana M. Kosovskaya, Doctor of Computer Science, Associate Professor;
Professor of Computer Science Chair of St.Petersburg State University
(SPbSU), 198504, Russia, Stary Petergof, St.Petersburg, University av., 28,
Faculty of Mathematics and Mechanics SPbSU; kosovtm@gmail.com**



Наши авторы, 2016.
Our authors, 2016.

Косовская Татьяна Матвеевна,
доктор физико-математических наук,
доцент, профессор кафедры информатики
математико-механического факультета
СПбГУ,
198504, Старый Петергоф, С.-Петербург,
Университетский пр., д. 28, мат-мех СПбГУ,
kosovtm@gmail.com