

# КОМПЬЮТЕРНЫЕ НАУКИ

УДК 004.822

## АЛГОРИТМ УНИФИКАЦИИ В МЕТОДЕ РЕЗОЛЮЦИЙ ДЛЯ ЛОГИКИ ПРЕДИКАТОВ *ПРОХОРОВ А. В., ПРОХОРОВ В. П.*

Рассматривается проблема повышения эффективности интеллектуальных систем, использующих в качестве механизма логического вывода метод резолюций для исчисления предикатов первого порядка. Показывается, что метод резолюций в процессе поиска (вывода) решений (следствий) основан на порождении большого числа резольвент и многократном использовании процедуры унификации. Предлагается новый алгоритм унификации, построенный на теории множеств и простых и единых правилах преобразования матриц возможных подстановок, что повышает его эффективность по сравнению с известными эвристическими процедурами. Данный алгоритм унификации может быть использован не только в методе резолюций, но и в других алгоритмах дедуктивного вывода, разработанных для логики предикатов.

### 1. Введение

В области искусственного интеллекта сегодня акценты сместились в сторону моделей машинного обучения, что позволило компьютерам стать умнее и решать более сложные задачи. Однако не следует забывать, что искусственный интеллект – это сложное поле для исследований, и оно состоит из нескольких компонентов, которые отвечают за общий прогресс в этой области. Так, в целом, всё что требует аргументации и рассуждений, находится пока за пределами возможностей моделей глубинного обучения, неважно, сколько данных в них загрузить. В этой связи исследования, связанные с разработкой интеллектуальных систем – различного рода решателей проблем – вопросно-ответных систем, экспертных систем, систем поддержки принятия решений – и, соответственно, совершенствования механизмов логического вывода, продолжают оставаться актуальными [1,2]. Проблема построения доказательства традиционно понимается как проблема поиска вывода формулы логико-математического языка в соответствующем исчислении. Имеются многочисленные работы, связанные со стандартными логиками (их разрешимыми фрагментами), а также рассматривающие неклассические исчисления (интуиционистские, нечеткие, дескриптивные и т.д.). В то же время предлагаемые формализации зачастую специфичны, не обладают универсальностью и не предлагают обобщений для классов (пусть и родственных) различных теорий представления и обработки знаний. Кроме того, далеко не всегда надлежащее внимание оказывается вопросам ал-

горитмической эффективности предлагаемых подходов.

Для построения подобного рода систем используется некоторый формальный язык и принцип дедуктивного вывода, что позволяет производить автоматический логический анализ сложившейся ситуации и делать верные логические выводы.

В большинстве случаев используется формализованный язык понятий и отношений, основанный на исчислении предикатов первого порядка, на котором формулируется описание предметной области, рекомендации и правила принятия решений (вывода следствий), которые представляются в виде аксиом.

Весь процесс принятия решения по оценке ситуаций, распознаванию объектов и их состояний, управлению объектами можно разбить на ряд последовательных этапов дедуктивного вывода. На каждом таком этапе алгоритм дедуктивного вывода решает вполне законченную задачу, состоящую в получении логических следствий из предложений, полученных на предыдущих этапах вывода из группы аксиом, строго соответствующей решаемой на данном этапе задаче логического вывода.

Таким образом, основу программного обеспечения автоматизированных систем, использующих интеллектуальные системы, должен составить алгоритм дедуктивного вывода (алгоритм доказательства теорем) следствий (доказательства теорем) на каждом из указанных выше этапов.

В работах [3,4] приводится достаточно полный обзор существующих методов и процедур доказательства. Наиболее эффективными из них являются обратный метод и его модификации, а также метод резолюций в сочетании с различными ограничительными стратегиями. Основными недостатками существующих алгоритмов и программ, основанных на известных методах поиска вывода, являются значительные затраты машинного времени и объема памяти при доказательстве даже простейших теорем.

Это, по-видимому, одна из причин, ограничивающих применение принципов искусственного интеллекта при построении адаптивных алгоритмов управления сложными системами, работающих в реальном масштабе времени.

Поэтому основным требованием при создании алгоритма дедуктивного вывода для сложных систем является его быстрдействие при доказательстве логических утверждений.

Создание новых процедур, позволяющих еще в большей степени, чем указанные методы, использовать идею локализации поиска по дереву вывода, а также введение эффективных эвристик в алгоритмы, основанные на существующих методах, является одним из путей преодоления возникающих трудностей. Одним из путей по-

вышения быстродействия алгоритмов, использующих принцип резолюций, является совершенствование процесса унификации при формировании резольвент.

В данной статье рассматриваются принципы построения нового алгоритма унификации, который является основополагающим при реализации дедуктивного вывода в формализованных теориях, построенных на основе исчисления предикатов первого порядка, в том числе и основанных на принципе резолюций.

## 2. Постановка задачи исследования

Процесс вывода некоторого утверждения (следствия) сводится к доказательству противоречия путем формирования общей унифицирующей подстановки. Так, Робинсон Дж. [4] для увеличения эффективности и расширения области практической применимости машинного (автоматического) доказательства теорем, базируясь на основополагающей теореме Эрбрана [3] для исчисления предикатов первого порядка, формулирует новый принцип вывода – принцип резольвенций (резолюций).

Этот метод доказательства объединил процессы подстановки (термов вместо переменных) и проверки результатов таких подстановок на тавтологичность. Доказана полнота принципа резольвенций. Однако процедура поиска доказательства, построенная непосредственно по теореме о резольвенции, неэффективна, так как процесс порождения резольвент может быть бесконечным, в силу теоремы Чёрча, хотя при этом используется лишь единственное правило вывода.

Дадим определение резольвенты. Пусть заданы исходные предложения в виде  $L = \{L_i\}$  и  $H = \{H_i\}$  и переменные, входящие в  $H = \{H_i\}$ , не встречаются в  $L = \{L_i\}$ . Пусть также  $\{l_i\} \subseteq \{L_i\}$  и  $\{h_i\} \subseteq \{H_i\}$  такие два подмножества в  $\{L_i\}$  и  $\{H_i\}$ , что для объединения  $\{l_i\} \cup \{h_i\}$  существует наиболее общий унификатор  $\lambda$ . Тогда два предложения  $L = \{L_i\}$  и  $H = \{H_i\}$  разрешаются, а новое предложение  $R = \{\{L_i\} - \{l_i\}\}_\lambda \cup \{\{H_i\} - \{h_i\}\}_\lambda$  является их резольвентой. Или иначе, если  $C$  и  $D$  – два дизъюнкта и  $L \subset C$ ,  $H \subset D$  – два одночленных множества, элементы которых образуют контрарную пару, то дизъюнкт  $R = (C - L) \cup (D - H)$  называется резольвентой  $C$  и  $D$ . Резольвента – это выведенное новое предложение, а процесс образования резольвенты из двух исходных предложений называется резольвенцией. При этом процесс резольвенции является общим правилом вывода, объединяющим подстановку *modus ponens* и

различные типы силлогизмов.

Согласно теореме о резольвенте [3] противоречие исходной системы  $SU \sim W$  доказано, если множество резольвент  $R^n$ , полученное на шаге  $n$  процесса последовательного порождения резольвент, пусто. Определение приведено для того, чтобы показать необходимость многократного применения процедуры унификации в процессе порождения новых резольвент на множестве литер. При этом исходную систему предложений можно представить в виде конечного множества литералов  $L = \{l_i\}$ , которые имеют сквозную нумерацию, начиная с первого предложения. На первом этапе из множества литералов  $L$  составляется множество контрарных пар  $X = \{X_{i,j}\}$ , где  $X_{i,j}$  – контрарная пара двух литералов  $l_i$  и  $l_j$ , причем, если  $l_i$  есть предикат  $P$ , то  $l_j = \neg P$ , и наоборот. Контрарная пара  $X_{i,j}$  образуется по правилам унификации для пары литералов.

Таким образом, одной из проблем реализации принципа резолюций в качестве правила вывода является необходимость многократного использования процедуры унификации литер, образующих контрарные пары. Следовательно, проблема разработки более эффективного алгоритма унификации актуальна.

## 3. Алгоритм унификации

Процесс унификации является основополагающим при реализации принципа дедуктивного вывода. Существует алгоритм унификации, который приводит шаг за шагом к наиболее общему унификатору для унифицируемого множества литералов  $L = \{l_i\}$ , если таковой существует.

Однако в алгоритме дедуктивного вывода, приведенном в [5, 6], алгоритм унификации отличен от подробно описанного в [3] эвристического алгоритма унификации. Рассмотрим новый алгоритм унификации, состоящий из двух основных этапов формирования матриц подстановок и их преобразования по простым правилам матричного анализа и теории множеств.

**Этап 1.** Формирование множества контрарных пар  $X$ . Первый этап по существу является подготовительным перед осуществлением полной унификации исходной системы предложений. Он реализован в специальном алгоритме унификации, с использованием которого формируются следующие множества в виде соответствующих квадратных матриц: контрарных пар  $X = \{X_{i,j}\}$ ; унифицирующих

подстановок  $\Pi = \{\Pi_{i,j}\}$ , где  $\Pi_{i,j}$  представляет собой множество разрешенных компонентов подстановки  $\Pi_{i,j} = \{m_{\alpha,\beta}\}$ ; запрещенных подстановок  $Z = \{Z_{i,j}\}$ , где  $Z_{i,j} = \{z_{\alpha,\beta}\}$  – множество запрещенных компонентов подстановки. Компоненты  $m_{\alpha,\beta}$  и  $z_{\alpha,\beta}$  определяются на множестве аргументов всех литер  $l_\gamma$ , принадлежащих исходной системе предложений  $L$ .

**Этап 2.** Формирование наиболее общей подготовки  $\Pi_0$ . Правило подстановки можно сформулировать в более общем виде: вместо предметной переменной  $y$ , принадлежащей некоторому классу понятий, разрешается подставить любые термы, принадлежащие данному классу, за исключением примитивных термов, представляющих собой функции Сколема от данной переменной [3]. Подстановки в другие термы являются запрещенными. Формирование общей подстановки  $\Pi_0$  есть процесс последовательного введения элементов  $\Pi_{i,j}$  множества  $\Pi$  в  $\Pi_0$ . Перед введением очередной подстановки  $\Pi_{i,j}$  производится проверка на противоречие подстановки  $\Pi_{i,j}$  элементам, принадлежащим  $\Pi_0$ . В случае противоречия происходит поиск другой  $\Pi_{i,j}$ , которая удовлетворяла бы условию непротиворечивости.

Покажем, как реализуется процесс выявления такого противоречия. Для этого  $\Pi_0$  представляется в виде симметричной булевой матрицы, описывающей отношение эквивалентности между отождествленными аргументами  $\Pi_0 = \|\|m_{\alpha,\beta}\|\|$ , столбцы и строки которой соответствуют аргументам множества  $L_0$ . Эквивалентность здесь понимается в том смысле, что предшествующие термы отождествляются с термами, следующими за ними в лексикографическом порядке. Элементы матрицы принимают лишь два значения: «истина» («1»), если в подстановку входит соответствующий компонент подстановки, и «ложь» («0») в противном случае.

Аналогично подстановкам  $\Pi_{i,j}$ , на первом этапе унификации формируется матрица постоянных запретов  $Z_0 = \|\|z_{\alpha,\beta}\|\|$ , где  $z_{\alpha,\beta}$  – компонент запрещенной подстановки, который принимает

значение «истина» («1») только в том случае, если подстановка  $\beta$ -го ( $\alpha$ -го) аргумента вместо  $\alpha$ -го ( $\beta$ -го) запрещена.

Введем еще одну матрицу  $A_0 = \|\|a_{\alpha,\beta}\|\|$ , которую назовем матрицей аргументов функций множества  $L_0$ ; так же как и матрицы  $\Pi_0$ ,  $Z_0$ , матрица  $A_0$  квадратная  $n \times n$ , где  $n$  – общее число аргументов множества  $L_0$ , но не симметричная. Элементы  $a_{\alpha,\beta}$  матрицы  $A_0$  принимают значение «1» лишь в том случае, если  $\alpha$  является аргументом функции  $\beta$ .

Тогда компоненты подстановки  $\Pi_{i,j}$  заносятся в матрицу  $\Pi_0$  и по формуле

$$A_1 = (\Pi_0^T \vee A_0)^T \cdot A_0 \cdot (\Pi_0^T \vee A_0)^T = (\Pi_0^T \cdot A_0 \cdot \Pi_0^T)^T \quad (1)$$

матрица аргументов функций  $A_0$  преобразуется в матрицу  $A_1$  с учетом подстановки  $\Pi_{i,j}$ , где  $t$  – означает операцию транзитивного замыкания матрицы. Умножение матриц производится по обычным правилам, но вместо операций *умножения и сложения* необходимо производить соответственно операции *конъюнкции и дизъюнкции*.

Затем с учетом  $\Pi_0$  и  $A_1$  формируется полная матрица запретов  $N$  на данном шаге по формуле  $N = (\Pi_0^T \cdot Z \vee Z \cdot \Pi_0^T) \vee A_1 \vee A_1^T = \Pi_0^T \cdot Z \cdot \Pi_0^T \vee A_1 \vee A_1^T$ , (2) где «1 – штрих» означает транспонирование матрицы.

Если в результате поэлементного сравнения матрицы  $N$  и  $\Pi_0^T$  по формуле

$$N \wedge \Pi_0^T = 0 \quad (3)$$

образуется нулевая матрица, то подстановка  $\Pi_{i,j}$  не противоречит  $\Pi_0$ . В противном случае обнаруживается противоречие. Проиллюстрируем сказанное на примере.

**Пример.** Пусть задана следующая система предложений, где термы (переменные, функции, константы) пронумерованы:

$$\begin{aligned} P_1(x_1, y_1) \vee P_2(y_1, z_1) : \\ : P_3(f(x_2, y_2), a), \end{aligned} \quad (4)$$

для которой на первом этапе унификации получены следующие подстановки:

$$\Pi_{1,3} = \{m_{1,6}, m_{2,7}\}, \quad \Pi_{2,3} = \{m_{2,6}, m_{3,7}\}$$

и запрещенная подстановка  $Z = \{z_{4,6}, z_{5,6}, z_{6,7}\}$ . Требуется получить общую подстановку  $\Pi_0$ , унифицирующую систему (4). Тогда должны

войти в  $\Pi_0$  – подстановки  $\Pi_{1,3}$  и  $\Pi_{2,3}$ , но в том случае, если  $\Pi_{1,3}$  не противоречат  $\Pi_{2,3}$ .

Запишем исходные матрицы:

$$\Pi_0 = \begin{pmatrix} 1000010 \\ 0100011 \\ 0010001 \\ 0001000 \\ 0000100 \\ 1100010 \\ 0110001 \end{pmatrix}, Z = \begin{pmatrix} 0000000 \\ 0000000 \\ 0000000 \\ 0000010 \\ 0000010 \\ 0001101 \\ 0000010 \end{pmatrix}, A_0 = \|0\|.$$

Согласно формулам (1) и (2) получим

$$\Pi_0^T = \begin{pmatrix} 1110011 \\ 1110011 \\ 1110011 \\ 0001000 \\ 0000100 \\ 1110011 \\ 1110011 \end{pmatrix}, A_1 = \begin{pmatrix} 0000000 \\ 0000000 \\ 0000000 \\ 1110011 \\ 1110011 \\ 0000000 \\ 0000000 \end{pmatrix},$$

$$H = \begin{pmatrix} 1111111 \\ 1111111 \\ 1111111 \\ 1110011 \\ 1110011 \\ 1111111 \\ 1111111 \end{pmatrix}.$$

В результате поэлементного сравнения матриц  $H$  и  $\Pi_0^T$  по формуле (3) получим непустую матрицу

$$H \wedge \Pi_0^T = \begin{pmatrix} 1110011 \\ 1110011 \\ 1110011 \\ 0000000 \\ 0000000 \\ 1110011 \\ 1110011 \end{pmatrix}.$$

Следовательно, образовать непротиворечивую общую подстановку в данном случае нельзя.

Заменим предикат  $P_2$  истемы предложений (4)

на предикат  $P_2 = (Z, y_1)$ , тогда изменится лишь подстановка  $\Pi_{2,3} = \{m_{2,7}, m_{3,6}\}$ , а матрицы  $Z$  и  $A_0$  останутся без изменения. В этом случае получим

$$\Pi_0^T = \begin{pmatrix} 1010010 \\ 0100001 \\ 1010010 \\ 0001000 \\ 0000100 \\ 1010010 \\ 0100001 \end{pmatrix}, A_1 = \begin{pmatrix} 0000000 \\ 0000000 \\ 0000000 \\ 1010010 \\ 1010010 \\ 0000000 \\ 0000000 \end{pmatrix},$$

$$H = \begin{pmatrix} 0101101 \\ 1010010 \\ 0101101 \\ 1010010 \\ 1010010 \\ 0101101 \\ 1010010 \end{pmatrix}, H \wedge \Pi_0^T = 0.$$

Таким образом, в данном случае образована непротиворечивая общая подстановка  $\Pi_0 = \{\Pi_{1,3}, \Pi_{2,3}\}$ , которая представляет собой наиболее общий унификатор для множества унифицируемых литералов (предикатов). При этом алгоритм унификации основан на реализации типовых операций в соответствии с новыми выражениями (1) и (2). Следует отметить, что эффективность метода резолюций зависит не только от процесса унификации, но и от выбора стратегий перебора возможных резольвент в процессе вывода, подробно рассмотренных в [3].

#### 4. Выводы

1. Применение классической логики предикатов при создании интеллектуальных систем обусловлено тем, что результаты вывода решений из системы предложений или нелогических аксиом формализованной теории обеспечивают наибольшую их достоверность при достаточно адекватном описании соответствующей предметной области. Это позволяет использовать их при решении важных трудноформализуемых логических и расчетно-логических задач в системах реального времени.
2. Для повышения производительности интеллектуальных систем, например, систем поддержки принятия решений, основанных на логике предикатов, впервые предложен новый алгоритм унификации.
3. Показано, что наиболее общая подстановка (наиболее общий унификатор) формируется на основании конкретных достаточно простых выражений теории множеств и преобразований квадратных матриц. Полученные результаты могут быть использованы при построении механизмов (алгоритмов) логического вывода в исчислении предикатов первого порядка, основанных на различных методах обработки знаний (метод резолюций, метод связанной конъюнкции, обратный метод), требующих

применения процесса унификации предикатов в аксиоматических формализованных теориях.

4. Данный алгоритм унификации позволяет расширить возможности разработки и применения интеллектуальных систем, основанных на классической логике предикатов, что обеспечит повышение качества, достоверности и сокращение времени на принятие решений в различных ситуациях и предметных областях.

Сказанное выше и определяет научную новизну и практическую значимость полученных результатов.

**Литература:** 1. Шайкин А.Н. Унификация в модели логического вывода / А.Н. Шайкин // Ученые записки Российского государственного социального университета. 2013.Т.1, № 5. С. 54-57. 2. Andréka, H. On A New Semantics for First-Order Predicate Logic / H. Andréka, J. van Benthem, I. Németi // Journal of Philosophical Logic. 2017.Vol. 46. P. 259-267. 3. Нильсон Н. Искусственный интеллект. Методы поиска решений [Текст] / Пер. с англ. В.Л. Стефанюка / Под ред. С.В. Фомина. М.: Мир, 1973. 272 с. 4. Robinson J. A. A Machine-Oriented Logic Based on the Resolution Principle, J. ACM, 12, No, 1, 23-41. [Русский перевод: см. Кибернетический сборник, №7, Новая серия, «Мир», М., 1970.] 5. Прохоров А. В. Облачная платформа разработки интеллектуальных систем поддержки принятия решений [Текст] / А. В. Прохоров, В. П. Прохоров, А. О. Матюшко // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. Х.: НТУ «ХПІ», 2015. №58 (1167). С. 72-76. 6. Прохоров В.П. Принятие решений в системе контроля и анализа параметров движения автотранспорта [Текст] / В.П. Прохоров, А.В. Прохоров // Радиоэлектроника и информатика. 2016.№1 (72). С.32-38.

#### **Transliterated bibliography:**

1. Shajkin A.N. Unifikacija v modeli logicheskogo vyvoda / A.N. Shajkin // Uchenye zapiski Rossijskogo gosudarstvennogo social'nogo universiteta. T.1, № 5. 2013. S. 54-57.  
2. Andréka, H. On A New Semantics for First-Order Predicate Logic / H. Andréka, J. van Benthem, I. Németi // Journal of Philosophical Logic. Vol. 46. 2017. pp. 259-267.  
3. Nil'son N. Iskusstvennyj intellekt. Metody poiska reshenij [Tekst] / Per. s angl. V.L. Stefanjuka / Pod red. S.V. Fomina. M.: Mir, 1973. 272 s.  
4. Robinson J. A. A Machine-Oriented Logic Based on the Resolution Principle, J. ACM, 12, No, 1, 23-41. [Russkij perevod: sm. Kiberneticheskij sbornik, №7, Novaja serija, «Mir», M., 1970.]

5. Prohorov A. V. Oblachnaja platforma razrabotki intellektual'nyh sistem podderzhki prinjatija reshenij [Tekst] / A. V. Prohorov, V. P. Prohorov, A. O. Matjushko // Visnik Nacional'nogo tehničnogo universitetu «HPI». Serija: Sistemnij analiz, upravlinnja ta informacijni tehnologii. 2015. H.: NTU «HPI», №58 (1167). S. 72-76.

6. Prohorov V.P. Prijnatie reshenij v sisteme kontrolja i analiza parametrov dvizhenija avtotransporta [Tekst] / V.P. Prohorov, A.V. Prohorov // Radiojelektronika i informatika. H.: NURE. №1 (72). 2016. С.32-38.

Поступила в редколлегию 02.12.2017

**Рецензент:** д-р техн. наук, проф. Филатов В.А.

**Прохоров Александр Валерьевич**, д-р техн. наук, доцент, профессор кафедры информационных управляющих систем Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ». Научные интересы: распределенные интеллектуальные системы, мультиагентные технологии, онтологии, Semantic Web, grid и cloud computing, промышленная автоматизация, моделирование систем. Адрес: Украина, 61070, Харьков, ул. Чкалова, 17, e-mail: [o.prokhorov@khai.edu](mailto:o.prokhorov@khai.edu).

**Прохоров Валерий Павлович**, канд. техн. наук, с.н.с., генеральный директор научного парка «Радиоэлектроника и информатика» ХНУРЭ. Научные интересы: интеллектуальные системы, системы поддержки принятия решений, механизмы логического вывода, формализованные теории, многозначные логики, категории и морфизмы. Адрес: Украина, 61166, Харьков, пр. Науки, 14, e-mail: [valery.prokhorov@nure.ua](mailto:valery.prokhorov@nure.ua).

**Prokhorov Alexander**, Doctor of Technical Sciences, Associate Professor, Professor of Information Control Systems Department, National Aerospace University "Kharkiv Aviation Institute". Research interests: distributed intelligent systems, multi-agent technology, ontology, Semantic Web, grid and cloud computing, industrial automation, systems modeling. Address: Ukraine, 61070, Kharkiv, Str. Chkalov, 17, e-mail: [o.prokhorov@khai.edu](mailto:o.prokhorov@khai.edu).

**Prokhorov Valery**, Candidate of Technical Sciences, Senior Research Fellow, CEO of the Science Park «Radio Electronics and Informatics», Kharkiv National University of Radio Electronics. Research interests: intelligent systems, decision support systems, automated reasoning, nonclassical logics, category theory. Address: Ukraine, 61166, Kharkiv, Science Avenue, 14, e-mail: [valery.prokhorov@nure.ua](mailto:valery.prokhorov@nure.ua).