

МЕТОД РЕЗОЛЮЦИЙ И СТРАТЕГИИ ПОИСКА ОПРОВЕРЖЕНИЙ

© 2021 М. В. Лещинская✉, Т. М. Леденева

*Воронежский государственный университет
Университетская пл., 1, 394018 Воронеж, Российская Федерация*

Аннотация. В статье предлагается новая стратегия управления выводом в методе резолюций для исчисления предикатов первого порядка. Данное исчисление является основой логической модели представления знаний, отличающейся высокой описательной мощностью. Используемое в исчислении предикатов понятие вывода на основе резолюции является максимально формализованным и эффективным. Логическая модель применима в основном в исследовательских системах, она предъявляет высокие требования и ограничения к предметной области. В статье приводятся основные сведения формальной теории предикатов первого порядка, описываются существующие стратегии и критерии, предлагаются новая стратегия и осуществляется ее сравнительный анализ с другими стратегиями. Ее отличительной особенностью является использование рейтингов, которые вычисляются для каждого дизъюнкта и учитывают его способность образовывать контрарную пару для порождения резольвенты. В статье приводятся алгоритмы, реализующие данную стратегию. Для сравнительного анализа стратегий используется ряд существующих стратегий и вводится новый критерий стратегии — количество дизъюнктов, многократно участвующих в построении резольвент.

Ключевые слова: исчисление предикатов, метод резолюций, стратегии управления выводом.

ВВЕДЕНИЕ

Интеллектуальные системы находят применение в различных предметных областях, опираясь на те или иные модели представления знаний. Логическая модель, базирующаяся на исчислении предикатов первого порядка, отличается высоким уровнем описательной способности, строгим теоретическим обоснованием правильности рассуждений, модульностью представляемых знаний, способностью к объяснению. Используемый в исчислении предикатов метод резолюций является максимально формализованным и эффективным инструментом доказательства теорем в исчислении предикатов [1–4]. Если предметная область включает детерминированные жесткие знания, то использование логической модели и алгоритмов логического

вывода становится целесообразным. Однако при создании, поддержке и эксплуатации больших баз знаний таких систем не всегда в полной мере учитывается объем ресурсов, необходимый для их функционирования. Практика показывает, что процесс работы с базами знаний, которые используют логические модели, является весьма ресурсозатратным [5]. Поэтому одним из основных требований при работе с логическим выводом в интеллектуальных системах является требование быстродействия [5, 6].

Метод резолюций (Дж. Робинсон, 1965 г.) основан на понятии резолютивного вывода и является одним из основных методов, которые используются для решения широкого круга задач [5]: для доказательства выводимости (доказуемости) формулы из множества посылок (гипотез); для выявления, является ли заданное множество формул противоречивым; для проверки правильности рассуждений. Исходной информацией для метода

✉ Лещинская Мария Владимировна
e-mail: maria-leshchinskaya@mail.ru



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.

является множество предложений — формул в виде элементарных дизъюнкций (дизъюнктов). Суть метода заключается в построении последовательности формул, которая заканчивается пустым дизъюнктом или выводом, что его получить нельзя. Построение данной последовательности является творческим процессом и в общем случае сводится к полному перебору, что снижает быстродействие метода в реальных приложениях. Попытки преодоления данного недостатка привели к появлению различных стратегий управления выводом [7–9]. Их цель — сократить полный перебор. И, если удастся угадать, что некоторая стратегия применима к имеющимся данным, то цель достижима. Другой подход заключается в совершенствовании алгоритма унификации [10]. Если удастся найти наиболее общий унификатор, то резольвенты можно упростить, а также сократить их количество, что является одним из путей повышения быстродействия алгоритмов, использующих принцип резолюций.

Основной алгоритм унификации также был предложен Дж. Робинсоном, но на сегодня не является наиболее эффективным из предложенных [9, 11–13]. В [10] упоминается алгоритм унификации, который обеспечивает неизменность главной операции формулы. При любых подстановках формул вместо переменных главная операция (предикат или функциональный символ) формулы не меняется. Поэтому, если главные операции формул различны, то формулы заведомо не унифицируемы тогда и только тогда, когда унифицируемы все подформулы, являющиеся аргументами главной операции.

В [3] описаны некоторые основные стратегии управления выводом (полный перебор, линейный алгоритм, стратегия опорного множества) и приведен их сравнительный анализ. В [5] представлены стратегия единичной резолюции, стратегия отрицательной резолюции, алгоритм входной резолюции. В общем случае, последние стратегии являются неполными. Если алгоритм метода резолюций с неполной стратегией не заканчивает свою работу, то это говорит только о том, что с помощью выбранной стратегии не найти

решение для рассматриваемой задачи. В [9] дается обзор различных стратегий для метода резолюций, описаны их отличительные черты, преимущества и недостатки. В [14] на конкретном примере рассматривает особенности логического вывода, основанного на логике предикатов.

Цель статьи заключается в представлении нового алгоритма для управления выводом в методе резолюций. Данный алгоритм основан на назначении дизъюнктам исходного множества рейтингов по специальным правилам, что позволяет уменьшить количество построенных резольвент и минимизировать количество дизъюнктов, участвующих в их построении.

1. МАТЕРИАЛЫ И МЕТОДЫ ИССЛЕДОВАНИЯ

1.1. Теоретическая база исследования

Введем основные теоретические понятия, базируясь на [3, 4].

Формальным исчислением называется кортеж

$$T = \langle \alpha, F, Axi, R \rangle,$$

где α — алфавит теории, F — множество правильно построенных формул, Axi — множество аксиом, R — множество правил вывода.

Важнейшим понятием всякого формального исчисления является понятие логического вывода, под которым подразумевается последовательность формул $\varphi_1, \dots, \varphi_n$, в которой любая формула является либо гипотезой, либо аксиомой, либо получена с помощью одного из правил вывода.

Рассмотрим конечное множество формул $\Gamma = \{H_1, \dots, H_m\}$, называемых гипотезами (посылками). Формула φ называется *следствием* множества формул Γ (а также выводимой, или доказуемой, формулой), и этот факт обозначается в виде $H_1, \dots, H_m \vdash \varphi$, если существует *логический вывод* формулы φ , т. е. такая последовательность формул $\varphi_1, \dots, \varphi_n$, в которой $\varphi_n = \varphi$, при этом для любого индекса $i = 1, n$ формула φ_i является либо гипотезой,

либо аксиомой теории T , либо непосредственным следствием предыдущих формул по одному из правил вывода из R . По сути, формула φ представляет собой новое знание, которое следует из гипотез H_1, \dots, H_n .

Для представления $H_1, \dots, H_n \mapsto \varphi$ можно рассматривать частные случаи. Если $\Gamma = \emptyset$, т. е. гипотезы H_1, \dots, H_n отсутствуют, то запись $\mapsto \varphi$ означает, что формула φ является теоремой, т. е. тождественно-истинной формулой. Запись $\varphi \mapsto$ означает, что формула φ является противоречием или тождественно-ложной формулой.

Легко увидеть связь между правильными рассуждениями и понятием выводимости формулы. Пусть H_1, \dots, H_n — гипотезы (предположения), φ — заключение. В исчислении предикатов первого порядка и гипотезы H_1, \dots, H_n , и формула φ должны представлять собой формулы логики предикатов.

Рассуждения являются *правильными* и этот факт обозначается в виде $\frac{H_1, \dots, H_n}{\varphi}$, если формула:

$$H_1 \wedge \dots \wedge H_n \rightarrow \varphi \equiv \bar{H}_1 \vee \dots \vee \bar{H}_n \vee \varphi$$

есть тавтология. Но тогда отрицание этой формулы $\bar{H}_1 \vee \dots \vee \bar{H}_n \vee \varphi \equiv H_1 \wedge \dots \wedge H_n \wedge \bar{\varphi}$ является противоречием. Таким образом, чтобы доказать, что формула φ выводима из множества формул Γ , нужно показать, что формула $H_1 \wedge \dots \wedge H_n \wedge \bar{\varphi}$ является противоречием. В этом случае множество формул $\{H_1, \dots, H_n, \bar{\varphi}\}$ называется *противоречивым*. Таким образом, задача, связанная с выводимостью формулы φ из множества гипотез H_1, \dots, H_n , сводится к тому, чтобы установить противоречивость множества формул $\{H_1, \dots, H_n, \bar{\varphi}\}$. Существует несколько способов для решения этой задачи [9]. В данной статье будем рассматривать *метод резолюций* [9]. Системы, основанные на резолюции, предназначены для создания доказательства (в форме логического вывода) путем выявления противоречия или, иными словами, построения *опровержения*. Его использование предполагает, что все формулы рассматриваемого множества представлены в форме элементарных дизъюнкций — *дизъюнктов*. В исчислении высказываний для этого достаточ-

но привести все формулы H_1, \dots, H_n и $\bar{\varphi}$ к конъюнктивной нормальной форме, а затем сформировать из дизъюнктов множество, перечислив их через запятую. В исчислении предикатов соответствующая процедура сложнее: необходимо вначале привести каждую формулу к предваренной нормальной форме, вынося все кванторы за скобку, затем избавиться от кванторов существования с помощью сколемовских функций и отбросить кванторы всеобщности. Результатом перечисления полученных дизъюнктов является множество предложений (клаузальное множество), с которым и работает метод резолюций.

Пусть H_1, \dots, H_n — гипотезы (предположения), φ — заключение. Чтобы доказать $H_1, \dots, H_n \mapsto \varphi$ методом резолюций необходимо выполнить следующие шаги [8]:

1. Приведение формул $H_1, \dots, H_n, \bar{\varphi}$ к нормальной форме, в которой присутствуют только операции \wedge, \vee, \neg .

2. Полученные формулы приводятся к предваренной нормальной форме, когда все кванторы выносятся за общую скобку.

3. Исключение кванторов существования за счет использования сколемовских функций и отбрасывание кванторов всеобщности.

4. Приведение к КНФ каждой формулы и формирование множества дизъюнктов S .

5. Разделение переменных таким образом, чтобы в каждом дизъюнкте присутствовали оригинальные переменные.

Метод резолюций в исчислении предикатов первого порядка позволяет доказать выводимость формулы не учитывая интерпретацию предикатных переменных. Последующая интерпретация всех составляющих формулы позволяет перейти к конкретным содержательным утверждениям и пополнить базу знаний, если метод применялся в рамках интеллектуальной системы.

Известно, что проблема разрешимости в исчислении предикатов первого порядка в общем случае не разрешима, т. е. не существует единой процедуры, позволяющей определить, является ли заданная формула теоремой или нет. Однако на основе априорного предположения об общезначимости форму-

лы, которую следует доказать, метод резолюций подтверждает, что формула общезначима (т.е. является теоремой), если она на самом деле таковой является [9]. Принцип резолюций обладает важным свойством — полнотой. Имеет место теорема [9]: множество дизъюнктов S противоречиво тогда и только тогда, когда существует вывод из S пустого дизъюнкта \square .

1.2. Метод резолюций

Под резолюцией будем понимать правило вывода, применение которого к двум формулам в форме дизъюнктов позволяет получить новую формулу как следствие [15, 16]. В исчислении высказываний под дизъюнктом понимается дизъюнкция переменных или их отрицаний. В исчислении предикатов дизъюнкт представляет собой дизъюнкцию атомов или их отрицаний, при этом атом — это предикатная переменная, за которой в круглых скобках следует список термов через запятую. Пропозициональную или предикатную переменную будем называть литералом, тогда дизъюнкт определим, как бескванторную дизъюнкцию литералов.

Основой для построения логического вывода является понятие резольвенты.

Пусть D_1 и D_2 — пара дизъюнктов, содержащих контрарные литералы A и \bar{A} . Это означает, что дизъюнкты представимы, например, в виде $D_1 = A \vee D'_1$ и $D_2 = \bar{A} \vee D'_2$. Резольвентой дизъюнктов D_1 и D_2 по переменной A называется дизъюнкт вида $res_A(D_1, D_2) = D'_1 \vee D'_2$. Для заданной пары дизъюнктов резольвента может существовать, причем не по одной переменной, если это возможно, а может и не существовать.

В случае исчисления предикатов определение более сложное.

Пусть D_1 и D_2 — дизъюнкты, не имеющие общих переменных, L_1 и L_2 — атомы, содержащиеся в этих дизъюнктах. Если L_1 и \bar{L}_2 имеют наиболее общий унификатор θ^* , то дизъюнкт

$$res(D_1, D_2) = (D_1\theta^* \setminus L_1\theta^*) \vee (D_2\theta^* \setminus \bar{L}_2\theta^*)$$

называется резольвентой D_1 и D_2 . Если D_1 и D_2 совпадают с атомами L_1 и L_2 соответственно, то $res(D_1, D_2) = \square$, где символом \square обозначен пустой дизъюнкт.

Пусть H_1, \dots, H_n — множество гипотез и φ — формула. Последовательность формул $\varphi_1, \dots, \varphi_n$ называется резольтивным выводом формулы φ из множества гипотез H_1, \dots, H_n , если $\varphi_n = \varphi$ и любая формула φ_i ($i = 1, n$) является или гипотезой, или аксиомой, или резольвентой каких либо предыдущих формул.

Имеет место следующее утверждение: если D_1 и D_2 — дизъюнкты и $res(D_1, D_2) = D$, то $D_1, D_2 \mapsto D$.

Как и в общем случае логического вывода цель заключается в получении пустого дизъюнкта.

Следует заметить, что в исчислении предикатов первого порядка нахождение контрарных пар литер затруднительно, поскольку список термов у предикатных переменных может различаться. И для устранения этого различия используется понятие наиболее общего унификатора.

Подстановка θ называется унификатором для множества формул F_1, \dots, F_k тогда и только тогда, когда $F_1\theta = \dots = F_k\theta$, где $F_j\theta$ есть результат применения подстановки к формуле F_j .

Унификатор σ называется наиболее общим унификатором (НОУ) для множества формул, если для каждого унификатора θ существует такая подстановка θ^* , что $\theta = \sigma \circ \theta^*$, где символом \circ обозначена композиция подстановок.

Итак, унификатор — это подстановка, при применении которой к заданным формулам, получается их общий случай, т. е. эти формулы превращаются в одну и ту же формулу. Очевидно, что унификаторов может быть несколько. Наиболее общий унификатор — это минимальная подстановка, которая унифицирует формулы.

Правило резолюции применимо только к предложениям, содержащим контрарный унифицируемые литералы, таким образом, резольвента — это дизъюнктивное соединение всех литералов за исключением пары контрарных.

Метод резолюций заключается в выполнении следующих шагов [9]:

Шаг 1. Если в S есть пустой дизъюнкт, то множество S не выполнимо, и алгоритм завершает свою работу, иначе перейти к следующему шагу.

Шаг 2. Найти в исходном множестве S такие дизъюнкты или склейки дизъюнктов D_1 и D_2 , которые содержат унифицируемые литералы $L_1 \in D_1$ и $L_2 \in D_2$. Если таких дизъюнктов нет, то исходное множество выполнимо и алгоритм завершает свою работу, иначе перейти к следующему шагу.

Шаг 3. Определить резольвенту $res(D_1, D_2)$, добавить ее в множество S и перейти к шагу 1.

Итак, для того чтобы применить метод резолюций, нужно найти контрарные литералы. В случае исчисления высказываний, такой проблемы нет. Любая переменная и та же переменная с отрицанием — это и есть пара контрарный литер. В случае исчисления предикатов первого порядка это не так просто. Чтобы определить резольвенту требуется предварительная унификация литералов, причем это не всегда возможно. В [10] описаны алгоритмы унификации, которые вначале проверяют, можно ли унифицировать заданную пару формул, и находят наиболее общий унификатор, если унификация возможна.

1.3. Стратегии поиска опровержений

1.3.1. Понятие стратегии и граф вывода

В силу неразрешимости логики предикатов первого порядка для выполнимого множества дизъюнктов S процедура, основанная на принципе резолюций, может работать бесконечно долго, реализуя полный перебор вариантов пар дизъюнктов. Однако за счет использования различных стратегий управления выводом полный перебор может быть значительно сокращен.

Введем в рассмотрение граф вывода [17]. Пусть $S = \{D_1, \dots, D_N\}$ — множество дизъюнктов (предложений), которое будем называть *базовым*. Каждому дизъюнкту поставим в соответствие вершину. Если два дизъюнкта D_i и D_j образуют резольвенту $D_{ij} = res(D_i, D_j)$,

то появляется новая вершина D_{ij} , которая соединяется ребрами с вершинами D_i и D_j . Для вершины D_{ij} вершины D_i и D_j являются родителями, а для вершин D_i и D_j вершина D_{ij} является их потомком. Заметим, что граф вывода, по сути, является упорядоченным двоичным деревом и может реализовать некоторую стратегию вывода. На первом уровне располагаются вершины, соответствующие дизъюнктам множества S . На каждой итерации формируется новый уровень, который содержит резольвенты для пар родителей, выбираемых по определенным правилам. Таким образом, резольвента каждого уровня, кроме первого, связана с двумя родителями, которые расположены на предыдущих уровнях. Если на некотором уровне получен пустой дизъюнкт \square , то построение графа вывода прекращается, и делается вывод, что исходное множество дизъюнктов невыполнимо (противоречиво). Если в процессе построения графа вывода оказывается, что никаких резольвент получить невозможно, то исходное множество не противоречиво.

К наиболее известным относят следующие стратегии [3]:

1) *стратегия полного перебора*, которая опирается на вычисление всевозможных резольвент каждого уровня;

2) *стратегия опорного множества*, опирающаяся на поиск опровержения с помощью опорного множества, то есть — множества, включающего дизъюнкты, полученные на основе отрицания цели, или их потомки;

3) *стратегия предпочтения одночленам*, при использовании которой в качестве родительской вершины пытается быть выбран одночлен;

4) *линейная стратегия*, которая для каждой резольвенты выделяет родителя (хотя бы одного), принадлежащего к базовому множеству.

1.3.2. Алгоритмы реализации стратегий

Изложим общий алгоритм вывода. В выводе будем использовать множество M , имеющее универсальный характер для описанных ограничительных стратегий. Потенци-

ально один из дизъюнктов для унификации всегда берется из множества M , поэтому важно корректно сформировать такое множество во время работы с определенной стратегией. Ниже упоминаются особенности инициализации множества M .

Шаг 1. Пусть k — счетчик итераций, S_k — множество резольвент на k -й итерации. Положить $k = 0$,

$$M = S \setminus S_k, S_{k+1} = \emptyset.$$

Шаг 2. Если во множестве S_k содержится пустой дизъюнкт \square , то множество S , и, следовательно, рассуждения являются правильными, алгоритм завершает свою работу, иначе — перейти к шагу 3.

Шаг 3. Если существуют дизъюнкты $D_i \in S_k$ и $D_j \in M$ такие, что из них можно получить резольвенту, то перейти к шагу 4, иначе — к шагу 5.

Шаг 4. Положить S_{k+1} :

$$S_{k+1} = S_k \cup \{res(D_i, D_j)\}$$

и перейти к шагу 3.

Шаг 5. Если $S_{k+1} = \emptyset$, то множество S выполнено, рассуждения не верны, и алгоритм завершает свою работу. Иначе положить $k = k + 1$, $S_{k+1} = \emptyset$, и перейти к шагу 2.

Действуя в соответствии с общим алгоритмом, известные стратегии управления выводом отличаются способом формирования множества M .

В случае *стратегии полного перебора множества M* совпадает с исходным множеством дизъюнктов S , а S_0 содержит произвольный дизъюнкт из S , из которого с помощью оставшихся дизъюнктов можно получить резольвенту. Сначала вычисляются все возможные резольвенты первого уровня, затем второго и так далее до уровня N . Резольвенты первого уровня — это резольвенты для исходного множества дизъюнктов M . Резольвенты i -го уровня — это резольвенты, у которых либо оба родителя берутся с $(i-1)$ -уровня, либо один родитель — с $(i-1)$ -уровня, а второй — с $(i-2)$ -уровня.

В случае *стратегии опорного множества* за M берется базовое множество S предложений без отрицания целевого, при этом S_k — отрицание целевого предложения или его по-

томки, а за S_0 — отрицание целевого предложения. Для вычисления резольвенты один из родителей выбирается из опорного множества.

В случае *стратегии предпочтения одночленам* за M берется множество одночленов, S_k — множество дизъюнктов на k -й итерации, а за S_0 — отрицание целевого предложения. Стратегия предпочтения одночленам является модификацией стратегии опорного множества, в которой вместо заполнения каждого уровня (как при полном переборе) выбирают одночлены в качестве родительской вершины для получения резольвенты. Такой процесс позволяет сконцентрировать поиск в направлении создания пустого дизъюнкта.

В случае *линейной стратегии* за M берется базовое множество дизъюнктов, S_k — множество резольвент на k -й итерации, S_0 — произвольный дизъюнкт из базового множества дизъюнктов, из которого с помощью оставшихся дизъюнктов можно получить резольвенту. Данная стратегия помогает выбрать одним из родителей элемент из базового множества в случае построения резольвенты.

Каждая из стратегий обладает своими недостатками и преимуществами [8]. Для сравнения стратегий используются различные критерии: количество уровней в дереве вывода, скорость роста множества резольвент, направление поиска, возможность реализации стратегии в зависимости от исходных данных, а также оцениваются полнота и простота.

Заранее определить, какой метод приведет к результату быстро и оптимально, невозможно. Однако, повысить эффективность решения задач с помощью метода резолюций можно за счет увеличения скорости и/или качества одной из частей (или всех частей), приводящих к результату: преобразование задачи в форму предложений, применение метода резолюций: выбор алгоритма унификации и стратегии поиска, осуществление доказательства, преобразование результата в понятный и подходящий для пользователя ответ.

2. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЯ

2.1. Описание алгоритмов, реализующих рейтинговую стратегию управления выводом

На основе анализа существующих стратегий вывода можно утверждать, что перспективным направлением совершенствования является учет всех преимуществ, которые характерны для рассмотренных выше стратегий, а именно предпочтение одночленов, использование опорного множества, резольвирование атомов из разных множеств и линейный подход к поиску опровержения. Новая стратегия основана на построении рейтинга предложений и позволяет проверять литералы на возможность унификации в определенном порядке. Она нацелена на предварительную работу с унифицируемыми литералами (именно такие литералы имеют наивысший рейтинг).

Пусть имеется базовое множество предложений. Каждому предложению присвоим рейтинг в соответствии со следующими правилами:

1) если атом имеет отрицание, то рейтинг предложения увеличить на 2, а если отрицание отсутствует, то на 1 (это позволит быстрее найти противоположные атомы и сократить время поиска унифицируемых литералов);

2) если предложение состоит из одного литерала, то его рейтинг увеличивается на 1 (данное правило реализует предпочтение одночленам, если они имеются, поскольку использование одночленов позволяет сократить полный перебор и прийти к пустому дизъюнкту и или к множеству предложений, для которого не существует резольвент);

3) если в предложении имеется часто встречающийся атом, то его рейтинг увеличивается на 1 (данное свойство дает возможность быстрее проверить на унификацию те литералы, что находятся в предложениях, состоящих из одинаковых атомов, так как просмотр предложений из одинаковых атомов может скорее привести к контрарным литералам).

Найденные на данной итерации рейтинги используются для формирования списка предложений в соответствии с убыванием рейтингов. Рейтинг предложений пересчитывается после каждой итерации, при этом под итерацией подразумевается единичный полный просмотр всего списка предложений и их возможная унификация. Цикл обновления рейтинга и унификации происходит до тех пор, пока не будет найден пустой дизъюнкт, или же пока не останется конечное множество предложений, состоящих из не унифицируемых литералов.

Такой метод поиска опровержения эвристически позволяет в некоторых задачах быстрее прийти к результату, сократить количество выводимых резольвент и уровней, за счет предварительного анализа предложений перед унификацией согласно вышеописанным свойствам.

Для изложения алгоритма введем следующие обозначения:

Пусть M — исходное множество дизъюнктов; P_t — t -предложение или t -дизъюнкт; R_t — рейтинг предложения t ; D_q — текущий q -атом, R_q — рейтинг атома q .

Алгоритм поиска

часто встречающихся дизъюнктов

Шаг 1. Взять $C \neq \emptyset$, состоящее из исходных атомов текущей задачи. Построить множество C'_1 из n элементов (где $n > 0$ — количество неповторяющихся атомов текущей задачи). Перейти к шагу 2.

Шаг 2. Построить множество C'_2 , в котором атомы из множества C'_1 расположены по убыванию встречаемости во множестве C . Перейти к шагу 3.

Шаг 3. Определить множество часто встречающихся элементов $C'' = \emptyset$. Положить $j = 0$. Перейти к шагу 3.1

Шаг 3.1 Добавить j -элемент множества $C''[j]$ во множество C'' и перейти к шагу 4.2.

Шаг 3.2 Если $j < n - 1$, то перейти к шагу 3.3, иначе к шагу 5.

Шаг 3.3 Если значение встречаемости j -элемента $C'_2[j]$ совпадает со значением встречаемости $j+1$ -элемента $C'_2[j+1]$, то по-

ложить $j = j + 1$ и перейти к шагу 3.1, иначе перейти к шагу 4.

Шаг 4. Алгоритм завершает свою работу. Построенное множество C'' состоит из часто встречающихся атомов текущей задачи.

Алгоритм построения рейтинга предложений

Шаг 1. Положить $R_t = 0, R_q = 0, t = 0, q = 0$.

Шаг 2. Если $D_q \in P_t$ содержит отрицание, то $R_t = R_t + 2$, и перейти к шагу 3.

Шаг 3. Если $D_q \in P_t$ единственный атом в предложении P_t , то $R_t = R_t + 1$. Перейти к шагу 4.

Шаг 4. Если $D_q \in M$ встречается во множестве C'' на текущей итерации (где C'' строится во время работы алгоритма поиска часто встречающихся дизъюнктов), то $R_t = R_t + 1$, где t — номер предложения, который содержит атом D_q . Перейти к шагу 5.

Шаг 5. Сформировать массив предложений по убыванию их рейтингов (данный массив представляет собой список предложения для получения резольвент и формируется на каждой итерации в методе резолюций).

Пусть M — исходное множество предложений, изменяющееся со временем построения резольвент; M_k — упорядоченное в соответствии с рейтингом множество M на k -й итерации; S_k — множество резольвент на k -й итерации.

Метод резолюций, основанный на построении рейтингов

Шаг 1. Положить $k = 0, M = S \setminus S_k, S_{k+1} = \emptyset$.

Шаг 2. Если во множестве S_k содержится пустой дизъюнкт \square , то исходное множество S предложений противоречиво, следовательно, рассуждения являются правильными, и алгоритм завершает свою работу. Иначе перейти к шагу 3.

Шаг 3. Если существуют дизъюнкты $D_i \in M$ и $D_j \in M$ такие, что из них можно получить резольвенту, то перейти к шагу 3.1, иначе — к шагу 5.

Шаг 3.1. Положить $i = 0, j = n - 1$ (где i, j — индексы элементов массива M_k , а n — количество предложений во множестве M).

Построить упорядоченное множество предложений M_k из M и перейти к шагу 3.2.

Шаг 3.2. Если существует резольвента для $M_k[i]$ и $M_k[j]$, то перейти к шагу 4, иначе перейти к шагу 3.3.1.

Шаг 3.3.1. Если $j > i$, положить $j = j - 1$ и перейти к шагу 3.2, иначе перейти к шагу 3.3.2.

Шаг 3.3.2. Если $i < n - 1$, положить $i = i + 1, j = n - 1$ и перейти к шагу 3.2, иначе перейти к шагу 5.

Шаг 4. Положить S_{k+1} :

$$S_{k+1} = S_k \cup \{res(D_i, D_j)\}$$

и перейти к шагу 5.

Шаг 5. Если $\square \in S_{k+1}$, то множество S невыполнимо (противоречиво), и алгоритм завершает свою работу. Иначе положить $k = k + 1, S_{k+1} = \emptyset$, и перейти к шагу 5.1.

Шаг 5.1. Если множество M не пусто, то изменить множество M по правилу: $M = M \setminus M_k[i] \cup M \setminus M_k[j], n = n - 2$, и перейти к шагу 2, иначе перейти к шагу 5.2.

Шаг 5.2. Алгоритм заканчивает свою работу. Если $\square \in S_{k+1}$, то множество S невыполнимо (противоречиво), в противном случае — однозначно сказать нельзя.

Во время работы алгоритма по построению рейтинга предложений на каждой итерации, получаем массив M_k дизъюнктов таких, что в начале последовательности располагаются предложения, состоящие по возможности из отрицательных часто встречающихся атомов или в принципе из одного атома, а в конце — дизъюнкты с атомами без отрицания. А значит, рассматривая первый и последний элементы массива, высока вероятность унификации таких дизъюнктов, ведь потенциально мы ищем атом и его отрицание. Таким образом, данный алгоритм позволяет сократить образование лишних дизъюнктов и множество резольвент, так как резольвированием потенциально унифицируемых дизъюнктов можно быстрее получить пустой дизъюнкт.

2.2. Иллюстративные примеры и сравнительный анализ

Пусть задано следующее множество предложений:

$$\{C(x) \vee O(x), Z(y) \vee \neg C(y), Y(A), \\ \neg Z(A), \neg O(z) \vee \neg Y(z)\}.$$

Докажем или опровергнем его противоречивость. Результаты работы алгоритма представлены в таблице ниже (табл. 1) и отображены на дереве вывода (рис. 1).

Далее продемонстрирована работа алгоритма полного перебора (рис. 2). Алгоритм завершает свою работу в случае получения уровня, на котором находится хотя бы один пустой дизъюнкт \square .

Для сравнения стратегий вывода применяются различные критерии [4, 5]:

- полнота стратегии;
- количество построенных резольвент;
- скорость роста дерева вывода;
- близость к двоичному дереву;
- реализуемость стратегии;
- количество уровней в дереве вывода;

Под *полной* подразумевается стратегия поиска опровержения, про которую известно, что если существует какое-либо опровержение по методу резолюции, то существует и опровержение по данной стратегии.

Под *скоростью роста* будем понимать коэффициент увеличения количества элементов во множестве построенных резольвент от уровня к уровню, при этом *общей скоростью роста* будем называть делитель количества резольвент, полученных на первом уровне, на значение количества резольвент, полученных на последнем уровне.

Такой критерий как, *количество уровней в дереве вывода*, демонстрирует скорость поиска опровержения — чем меньше уровней, тем алгоритм быстрее, так как решение находится за наименьшее число шагов (глобальных итераций).

Анализ различных стратегий обосновывает использование нового критерия — количество дизъюнктов, многократно участвующих в построении резольвент. Наименьшее количество действий унификации ускоряет работу алгоритма, позволяя не тратить время на попытки построения резольвент между множественными дизъюнктами, а также упрощает дерево вывода, позволяя быстрее анализировать поиск решения.

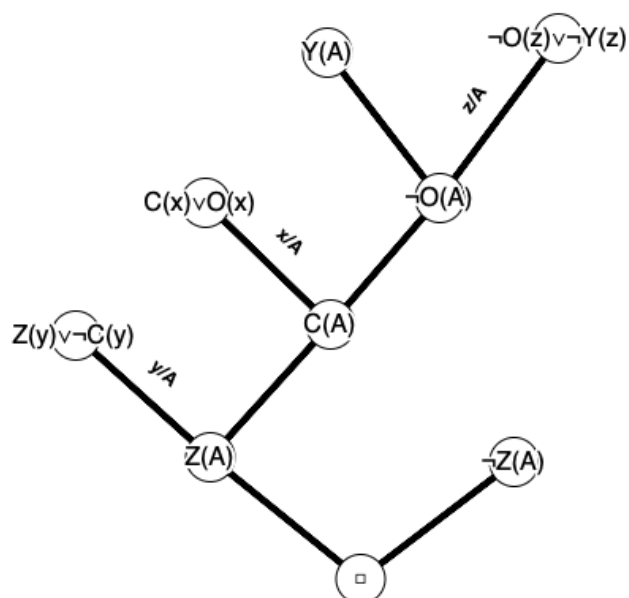


Рис. 1. Дерево вывода, соответствующее алгоритму с рейтингами
[Fig. 1. The inference tree which corresponds to the search algorithm with rating]

С учетом перечисленных критериев стратегия, основанная на построении рейтинга, имеет следующие преимущества по сравнению со стратегией полного перебора:

– количество резольвент для рейтинговой стратегии значительно меньше, чем в случае стратегии полного перебора;

– количество дизъюнктов, многократно участвующих в построении резольвент, также больше в случае алгоритма полного перебора, о чем наглядно свидетельствует дерево вывода (рис. 2);

– дерево вывода для стратегии, основанной на построении рейтингов, имеет больше уровней чем дерево, которое соответствует полному перебору (рис. 1), но в первом случае имеем упорядоченное двоичное дерево, которое обладает высоким уровнем объяснительной способности.

Анализ проведенных вычислительных экспериментов подтверждает приведенные выше выводы. В подавляющем большинстве случаев стратегия, основанная на построении рейтинга предложений, является более предпочтительной, чем известные классические стратегии.

Таблица 1. Результаты работы алгоритма на основе построения рейтинга
 [Table 1. The results of the algorithm based on the construction of the rating]

Номер текущего шага	Результат	Номер следующего шага
1	$k = 0,$ $M = \{C(x) \vee O(x), Z(y) \vee \neg C(y), Y(A), \neg Z(A), \neg O(z) \vee \neg Y(z)\},$ $S_0 = \emptyset$	2
2	S_0 не содержит пустого дизъюнкта	3
3	В M есть унифицируемые литералы	3.1
3.1	$R(C(x) \vee O(x)) = 1 + 1 + 1 = 3,$ $R(Z(y) \vee \neg C(y)) = 1 + 2 + 1 = 4,$ $R(Y(A)) = 1 + 1 + 1 = 3,$ $R(\neg Z(A)) = 2 + 1 + 1 = 4,$ $R(\neg O(z) \vee \neg Y(z)) = 2 + 2 + 1 = 5,$ $M_0 = \{\neg O(z) \vee \neg Y(z), Z(y) \vee \neg C(y), \neg Z(A), C(x) \vee O(x), Y(A)\}$	3.2
3.2	$D_i = M_0[1] = \neg O(z) \vee \neg Y(z),$ $D_j = M_0[n-1] = Y(A)$	4
4	$S_1 = \{\neg O(z)\}$	5
5	$k = 1, S_1 = \emptyset$	2
2	S_1 не содержит пустого дизъюнкта	3
3	В M есть унифицируемые литералы	3.1
3.1	$R(\neg O(z)) = 2 + 1 = 3,$ $R(Z(y) \vee \neg C(y)) = 1 + 2 = 3,$ $R(\neg Z(A)) = 2 + 1 = 3,$ $R(C(x) \vee O(x)) = 1 + 1 = 2,$ $M_1 = \{\neg O(z), Z(y) \vee \neg C(y), \neg Z(A), C(x) \vee O(x)\}$	3.2
3.2	$D_i = M_1[1] = \neg O(z),$ $D_j = M_1[n-1] = C(x) \vee O(x)$	4
4	$S_2 = \{\neg O(z), C(A)\}$	5
5	$k = 2, S_2 = \emptyset$	2
2	S_2 не содержит пустого дизъюнкта	3
3	В M есть унифицируемые литералы	3.1
3.1	$R(C(x)) = 1 + 1 = 2,$ $R(Z(y) \vee \neg C(y)) = 1 + 2 = 3,$ $R(\neg Z(A)) = 2 + 1 = 3,$ $M_2 = \{Z(y) \vee \neg C(y), \neg Z(A), C(x)\}$	3.2
3.2	$D_i = M_2[1] = Z(y) \vee \neg C(y),$ $D_j = M_2[n-1] = C(x)$	4
4	$S_3 = \{\neg O(z), C(A), Z(x)\}$	5
5	$k = 3, S_3 = \emptyset$	2
2	S_3 не содержит пустого дизъюнкта	3
3	В M есть унифицируемые литералы	3.1
3.1	$R(Z(x)) = 1 + 1 = 2,$ $R(\neg Z(A)) = 2 + 1 = 3,$ $M_3 = \{Z(x), \neg Z(A)\}$	3.2
3.2	$D_i = M_3[1] = Z(x),$ $D_j = M_3[n-1] = \neg Z(A)$	4
4	$S_3 = \{\neg O(z), C(A), Z(z), \square\}$	5
5	S_3 содержит пустой дизъюнкт, алгоритм завершает работу	

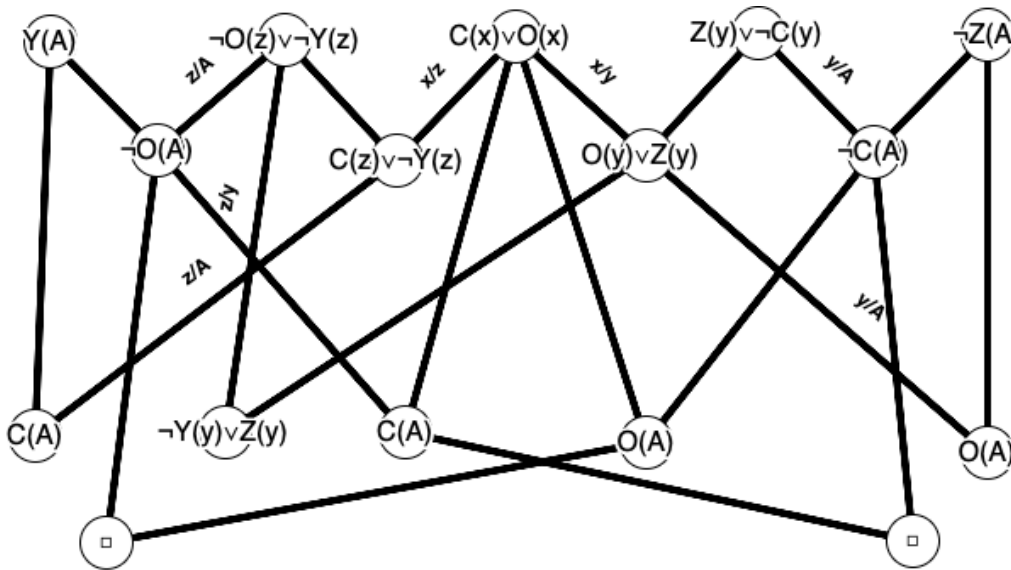


Рис. 2. Дерево вывода, соответствующее алгоритму полного перебора
 [Fig. 2. The inference tree which corresponds to the full search algorithm]

ЗАКЛЮЧЕНИЕ

В статье описан метод резолюций, а также приведены варианты стратегий для построения дерева вывода. Предложена новая стратегия, основанная на эвристических соображениях, которая позволяет сформировать список для просмотра дизъюнктов с целью минимизации количества операций, необходимых для унификации дизъюнктов при построении резольвенты. Предложенный алгоритм позволяет уточнить направление поиска в сторону пустого дизъюнкта и минимизировать количество дизъюнктов, участвующих в построении резольвент, что позволит ускорить алгоритм поиска решения.

В дальнейшем планируется проанализировать предложенную стратегию на полноту, убедиться в её эффективности, затем рассмотреть особенности компьютерной реализации существующих и предложенного решений, а также экспериментально получить и исследовать результаты.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Ахо, А. Теория синтаксического анализа перевода и компиляции / А. Ахо, Дж. Ульман. – М. : Мир, 1978.
2. Чень, Ч. Математическая логика и автоматическое доказательство теорем / Ч. Чень, Р. Ли. – М. : Наука, 1983.
3. Логический подход к искусственному интеллекту. Т. 1. От классической логики к логическому программированию / Тейз А. [и др.]. – М. : Мир, 1990.
4. Baader, F. The Description Logic Handbook / F. Baader. – New York : Cambridge University Press, 2003.
5. Нильсон, Н. Искусственный интеллект. Методы поиска решений. Пер. с англ. Стефанюка В. Л. / Под ред. Фомина С. В. – М. : Мир, 1973. – 272 с.
6. Маслов, С. Ю. Обратный метод установления выводимости в классическом исчислении предикатов / С. Ю. Маслов // ДАН СССР. – 1964. – Вып. 159, № 1. – С. 17–20.
7. Леденева, Т. М. Формальные аксиоматические теории. Исчисление предикатов. Часть 1 / Т. М. Леденева, Е. М. Аристова. – Воронеж : Издательский дом ВГУ, 2016.
8. Леденева, Т. М. Формальные аксиоматические теории. Исчисление предикатов. Часть 2 /

Т. М. Леденева. – Воронеж : Издательский дом ВГУ, 2020.

9. *Поспелов, Д. А.* Искусственный интеллект. – В 3-х кн. Кн. 2. Модели и методы: Справочник / Под ред. Д. А. Поспелова. – М. : Радио и связь, 1990. – 304 с.

10. *Новиков, Ф. А.* Системы представления знаний / Ф. А. Новиков // Санкт-Петербург : Издательство Политехнического университета, 2011.

11. *Sowyer, B.* Programming Expert Systems in Pascal / B. Sowyer, D. Foster. – John Wiley & Sons, Inc, 1986. – 198 p.

12. *Maciol, A.* An application of rule-based tool in attributive logic for business rules modeling / A. Maciol // Expert Systems with Applications. – April 2008. – V. 34, No 3. – P. 1825–1836.

13. *Robinson, J. A.* A Machine-Oriented Logic Based on the Resolution Principle / J. A. Robinson // J. ACM. – No 12.

14. *Тюрин, С. Ф.* Особенности логического вывода в пролог-программах / С. Ф. Тюрин, А. Ю. Городилов // Вестник Пермского университета. Серия: Математика. Механика. Информатика. – 2019. – № 3(46). – С. 91–97.

15. *Slagle, J.* Automatic theorem proving with renamable and semantic resolution / J. Slagle // Journal of the Association for Computing Machinery. – 1967. – V. 14. – P. 687–697.

16. *Братчиков, И. Л.* Формально-грамматическая интерпретация логического вывода / И. Л. Братчиков, Н. В. Сазонова // Санкт-Петербург, СПГЭУ «ЛЭТИ» им. В. И. Ульянова (Ленина). – Компьютерные инструменты в образовании, н. 6. – 2010. – С. 31–40.

17. *Касьянов, В. Н.* Графы в программировании: обработка, визуализация и применение / В. Н. Касьянов, В. А. Евстигнеев. – СПб. : БХВ-Петербург, 2003. – 1104 с.

Леденева Татьяна Михайловна — д-р техн. наук, профессор, заведующий кафедрой вычислительной математики и прикладных информационных технологий, факультет прикладной математики, информатики и механики, Воронежский государственный университет.

E-mail: ledeneva-tm@yandex.ru

ORCID iD: <https://orcid.org/0000-0002-3944-2266>

Лещинская Мария Владимировна — аспирант 2 года обучения, факультет прикладной математики, информатики и механики, Воронежский государственный университет.

E-mail: maria-leshchinskaya@mail.ru

ORCID iD: <https://orcid.org/0000-0001-6944-8485>

THE RESOLUTION METHOD AND REFUTATION STRATEGIES

© 2021 M. V. Leshchinskaya✉, T. M. Ledeneva

Voronezh State University
1, Universitetskaya Square, 394018 Voronezh, Russian Federation

Annotation. The article suggests a new inference control strategy to be used in the resolution method for first-order logic. This calculus is the basis of a logical model of knowledge representation, which is characterized by its high descriptive power. The notion of resolution inference used in predicate calculus is as formalised and efficient as possible. The logical model is mainly applicable to research systems, because it imposes high requirements and restrictions on the subject area. In the article, we discuss the basics of the formal theory of first-order logic and describe the existing strategies and criteria. We also suggest a new strategy and compare it with the existing ones. A distinctive feature of this strategy is that it uses ratings, which are calculated for each disjunction. The ratings take into account the ability of each disjunction to form a complementary pair to generate a resolvent. The article also describes the algorithms which implement the suggested strategy and recommends a new strategy criterion — the number of disjunctions which repeatedly participate in the generation of resolvents.

Keywords: first-order logic, resolution inference, inference control strategies.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. Aho A., Ullman J. (1978) The theory of parsing, translation and compiling. Moscow : Mir.
2. Chen Ch., Li R. (1983) The mathematical logic and automated theorem proving. Moscow : Science.
3. Teise A. [et al.] (1990) A logical approach to artificial intelligence. T.1. From classical logic to logic programming. Moscow : Mir.
4. Baader F. (2003) The Description Logic Handbook. New York : Press of the Cambridge University.
5. Nilsson N. (1973) Artificial intelligence. Methods for finding solutions. Per. from English Stefanuk V. L. Ed. Fomina S. V. Moscow : Mir. 272 p.
6. Maslov S. Yu. (1964) Backward method for establishing deducibility in the classical predicate calculus. SSS. Iss. 159, No 1. P. 17–20.
7. Ledeneva T. M., Aristova E. M. (2016) The formal axiomatic theories. Calculus of predicates. Part 1. Voronezh : Publishing House of the Voronezh State University.
8. Ledeneva T. M. (2020) The formal axiomatic theories. Calculus of predicates. Part 2. Voronezh : Publishing House of the Voronezh State University.
9. Pospelov D. A. (1990) Artificial intelligence. In 3 books. Book. 2. Models and Methods: Handbook. Ed. D. A. Pospelova. Moscow : Radio and communication. 304 p.
10. Novikov F. A. (2011) The systems of knowledge representation. St. Petersburg : Publishing House of the Polytechnic University.
11. Sowyer B., Foster D. (1986) Programming Expert Systems in Pascal. John Wiley & Sons, Inc. 198 p.
12. Maciol A. (2008) An application of rule-based tool in attributive logic for business rules modeling // Expert Systems with Applications. April 2008. Vol. 34, No. 3. P. 1825–1836.
13. Robinson J. A. A Machine-Oriented Logic Based on the Resolution Principle // J. ACM. No 12.
14. Tyurin S. F., Gorodilov A. Yu. (2019) The features of logical inference in prolog programs //

✉ Leshchinskaya Maria V.
e-mail: maria-leshchinskaya@mail.ru

Bulletin of Perm University. Series: Mathematics. Mechanics. Computer science. No 3 (46). P. 91–97.

15. *Slagle J.* (1967) Automatic theorem proving with renamable and semantic resolution // Journal of the Association for Computing Machinery. Vol. 14. P. 687–697.

16. *Bratchikov I. L., Sazonova N. V.* (2010) The formal-grammatical interpretation of logi-

cal inference. St. Petersburg, St. Petersburg Electrotechnical University “LETI” them. V. I. Ulyanov (Lenin). Computer tools in education, n. 6. P. 31–40.

17. *Kasyanov V. N., Evstigneev V. A.* (2003) The graphs in programming: processing, visualization and application. SPb. : BHV-Petersburg. 1104 p.

Ledeneva Tatyana M. — DSc in Technical Sciences, Professor, Head of the Department of Computational Mathematics and Applied Information Technologies, Faculty of Applied Mathematics, Informatics, and Mechanics, Voronezh State University.

E-mail: ledeneva-tm@yandex.ru

ORCID iD: <https://orcid.org/0000-0002-3944-2266>

Leshchinskaya Maria V. — 2nd year postgraduate student, Faculty of Applied Mathematics, Informatics, and Mechanics, Voronezh State University.

E-mail: maria-leshchinskaya@mail.ru

ORCID iD: <https://orcid.org/0000-0001-6944-8485>