

Алгоритмы релевантного обратного вывода на основе решения продукционно-логических уравнений

Аннотация. Представлены алгоритмы обратного вывода для продукционных систем, основанные на решении уравнений в LP-структурах – специальном классе алгебраических систем. Стратегия релевантного вывода направлена на оптимизацию обращений к внешнему источнику информации (базе данных или интерактивному пользователю). Концепция уравнений также применима для верификации знаний. Преимущества новых алгоритмов экспериментально демонстрируются на тестовых базах знаний.

Ключевые слова: решетка, логическое уравнение, продукционная система, обратный вывод, верификация знаний.

Введение

При решении задач представления знаний и управления знаниями достаточно эффективными оказываются алгебраические методы [1–2].

В работах [3–4] предложен математический аппарат, позволяющий рассматривать формализацию знаний и моделирование продукционно-логического вывода с точки зрения теории решеток и отношений. Согласно данному подходу, информация о предметной области хранится в виде иерархической структуры – решетки. Основная идея состоит в представлении продукционных связей (то есть совокупности правил) дополнительным бинарным отношением с заданными свойствами (рефлексивность, транзитивность и некоторые другие свойства, зависящие от конкретной модели). При этом определяющее решетку исходное отношение частичного порядка отражает универсальные тавтологии и является фиксированным. Дополнительное отношение порождается логическими связями конкретной предметной области и может подвергаться эквивалентным преобразованиям. Решетка с заданным на ней логическим отношением была названа LP-структурой (Lattice Production Structure).

Для введенного класса логических отношений был изучен ряд основных вопросов: замыкание, эквивалентные преобразования, каноническая форма, логическая редукция. Эти результаты мо-

гут быть применены для автоматизированных формальных исследований баз знаний и логических программ, включая верификацию и оптимизацию. В статье [4] задача обратного продукционно-логического вывода сведена к решению уравнений в LP-структурах. Рассмотрены вопросы о разрешимости и количестве решений этих уравнений, а также способы их решения.

В докладе [5] изложены идеи и некоторые результаты практического применения таких уравнений для ускорения обратного вывода и верификации знаний. Стратегия релевантного вывода направлена на минимизацию количества медленно выполняемых запросов (к базе данных или интерактивному пользователю). Запросы по возможности соответствуют фактам, которые действительно необходимы при выводе. Отрицательный ответ на запрос исключает последующие запросы об элементах связанного подмножества фактов. Кроме того, при LP-выводе предпочтение отдается тестированию множеств фактов минимальной мощности.

В настоящей работе впервые представлены формальная постановка задачи и соответствующие алгоритмы для метода усовершенствованного обратного вывода, основанные на указанных концепциях.

Предложенная методология не пересекается с известными стратегиями быстрого вывода, а дополняет их. Во-первых, алгоритмы RETE [6] и TREAT [7], базирующиеся на специальных

сетевых представлениях множеств правил, изначально разрабатывались для *прямого* вывода и использовались в продукционных системах с прямым выводом (например, OPS5, CLIPS). Алгоритм LEAPS [8] осуществляет компиляцию в язык С множества правил той же продукционной системы OPS5. В последующих исследованиях алгоритм RETE адаптировался для смешанного (двунаправленного) вывода [9]. Наши методы предназначены исключительно для *обратного* вывода и поэтому не требуют громоздких динамических структур данных, свойственных известным алгоритмам, в случаях, когда нет потребности в прямом (или смешанном) выводе.

Статья состоит из следующих основных разделов. В п.1 вводятся необходимые базовые понятия и обозначения; затем кратко приводятся формальные результаты работ [3–4], на которых основано дальнейшее исследование. В п.2 изложенные теоретические концепции применяются для формализации экспертных продукционных систем известного класса [10]. В п.3 излагается метод, дается формальная постановка задачи релевантного вывода, а также приводятся алгоритмы ее решения. В п.4 демонстрируются преимущества рассмотренной методологии на тестах, включающих пример базы знаний из книги [10]. В заключении подводятся итоги исследования и указываются некоторые перспективы.

1. Базовые понятия и теоретические результаты

Напомним вначале некоторые определения из теории отношений и решеток.

Бинарное отношение R на произвольном множестве F называется транзитивным, если для любых $a, b, c \in F$ из $(a, b), (b, c) \in R$ следует $(a, c) \in R$. Известно [11], что существует замыкание R^* произвольного отношения R относительно свойства транзитивности (транзитивное замыкание). Обратная задача – нахождение транзитивной редукции: по данному R ищется минимальное отношение R' такое, что его транзитивное замыкание совпадает с транзитивным замыканием R . Напомним также, что для частично упорядоченных множеств различаются понятия минимального элемента (для него нет меньшего элемента) и наименьшего

элемента (он меньше всех) [12]. В [11] приведен алгоритм построения транзитивной редукции; показано, что эта задача вычислительно эквивалентна построению транзитивного замыкания, и доказана единственность транзитивной редукции ациклического графа.

Решеткой [11] называется множество с частичным порядком $\leq$ («не больше», «содержится»), на котором для любой пары элементов определены операции $\wedge$ («пересечение») и $\vee$ («объединение»). Элемент $c = a \wedge b$ – это точная нижняя грань элементов a, b , то есть наибольший элемент решетки, удовлетворяющий неравенствам $c \leq a$ и $c \leq b$. Соответственно $d = a \vee b$ – точная верхняя грань a, b , то есть наименьший элемент решетки, для которого выполнено $a \leq d$ и $b \leq d$.

Решетка $\mathbb{F}$ называется ограниченной, если она содержит наименьший элемент O и наибольший элемент I , то есть $O \leq a \leq I$ для любого $a \in \mathbb{F}$.

В настоящей работе в качестве основы LP-структур используется булеан (множество всех подмножеств некоторого универсума) – частный случай ограниченной решетки. Поэтому вместо символов $\leq$, $\geq$, $\wedge$ и $\vee$ используются знаки теоретико-множественных операций $\subseteq$, $\supseteq$, $\cap$ и $\cup$, а элементы решетки обычно обозначаются большими буквами.

Атомом ограниченной (снизу) решетки $\mathbb{F}$ называется минимальный элемент ее подмножества $\mathbb{F} \setminus O$. Атомы будем обозначать маленькими буквами. Решетка называется атомно-порожденной, если каждый ее элемент может быть представлен объединением атомов. Например, в булеане атомами являются все подмножества, состоящие ровно из одного элемента базового множества.

Бинарное отношение R на решетке называется дистрибутивным [3], если из $(A, B_1), (A, B_2) \in R$ следует $(A, B_1 \cup B_2) \in R$. Отношение называется логическим, если оно содержит $\supseteq$, дистрибутивно и транзитивно. Логическим замыканием $\xrightarrow{R}$ произвольного отношения R называется наименьшее логическое отношение, содержащее R .

Под LP-структурой подразумевается решетка с заданным на ней логическим отношением. Такие алгебраические структуры могут служить основой для формализации и исследова-

ния широко распространенных на практике логических систем продукционного типа.

В работе [3] изучены основные свойства LP-структур данного класса. В частности, доказана теорема о существовании логического замыкания произвольного бинарного отношения на решетке, что позволяет ввести понятие эквивалентного отношения. Два отношения, заданные на общей решетке, называются эквивалентными, если их логические замыкания совпадают.

Доказаны также теоремы о возможностях локально-эквивалентных преобразований исходного отношения. Эти результаты обосновывают формальные преобразования знаний продукционного типа.

Отношение R на атомно-порожденной решетке $\mathbb{F}$ называется каноническим, если оно задано множеством пар вида (A, a) , где $A \in \mathbb{F}$, a – атом в $\mathbb{F}$. В [3] доказана теорема о существовании эквивалентного канонического представления для произвольного отношения.

Рассмотрен вопрос о минимизации отношений с сохранением их свойств. Логической редукцией отношения R на решетке называется любое минимальное отношение, эквивалентное R . Доказана теорема о существовании логической редукции и описан способ ее построения, что позволяет формулировать и успешно решать задачи автоматической оптимизации соответствующих формальных знаний.

Наконец, в работе [4] введен новый класс продукционно-логических уравнений и обоснован метод их решения, что в применении к продукционным системам соответствует полному обратному выводу. На уравнениях остановимся немного подробнее, поскольку они играют основную роль в качестве теоретической основы настоящей статьи.

Пусть дано бинарное отношение R на решетке $\mathbb{F}$ и имеет место $A \xrightarrow{R} B$. Тогда в смысле общей теории B называется образом A , а A – прообразом B при отношении $\xrightarrow{R}$. Поскольку отношение R произвольно, то каждый элемент из $\mathbb{F}$ может иметь много образов и прообразов. Более того, в случае логического отношения $\xrightarrow{R}$ любой $B_1 \subset B$ является образом A и каждый $A_1 \supset A$ является прообразом B . Поэтому требуется уточнение рассматриваемых понятий.

Для элемента $B \in \mathbb{F}$ минимальным прообразом при отношении $\xrightarrow{R}$ называется такой

элемент $A \in \mathbb{F}$, что $A \xrightarrow{R} B$ и A является минимальным, то есть не содержит никакого другого $A_1 \in \mathbb{F}$, для которого $A_1 \xrightarrow{R} B$.

Определение 1. Атом x решетки $\mathbb{F}$ называется начальным при отношении R , если в R нет ни одной такой пары (A, B) , что x содержится в B и не содержится в A . Элемент X атомно-порожденной решетки $\mathbb{F}$ называется начальным, если все его атомы являются начальными. Подмножество $\mathbb{F}_0(R)$ (будем писать $\mathbb{F}_0$, если это не вызовет неоднозначностей) решетки $\mathbb{F}$, состоящее из всех начальных элементов $\mathbb{F}$, называется начальным множеством решетки $\mathbb{F}$ (при отношении R).

Рассмотрим уравнение

$$R(X) = B, \quad (1)$$

где $B \in \mathbb{F}$ – заданный элемент, $X \in \mathbb{F}$ – неизвестный.

Определение 2. Частным решением X уравнения (1) называется любой минимальный прообраз элемента B , содержащийся в $\mathbb{F}_0$. Приближенным (частным) решением X уравнения (1) называется любой прообраз элемента B , содержащийся в $\mathbb{F}_0$. Общим решением уравнения (1) называется совокупность всех его частных решений $\{X_s\}$, $s \in S$.

Уравнения вида (1) будем называть продукционно-логическими уравнениями на решетках.

Рассмотрим продукционно-логическое уравнение следующего вида:

$$R(X) = B_1 \cup B_2 \quad (2)$$

Теорема 1. Пусть $\{X_s\}$, $s \in S_1$ – общее решение уравнения вида (1) с правой частью B_1 , а $\{Y_p\}$, $p \in S_2$ – общее решение уравнения того же вида с правой частью B_2 . Тогда общее решение уравнения (2) представляет собой множество всех элементов вида $X_s \cup Y_p$, из которого исключены элементы, содержащие другие элементы этого же множества.

Предположим, что R является каноническим отношением на атомно-порожденной решетке, а правая часть уравнения (1) представляет собой конечное объединение атомов. Тогда, в силу теоремы 1, вместо (1) достаточно рассмотреть уравнения с атомами в правой части:

$$R(X) = b \quad (3)$$

В работе [4] сформулированы и доказаны условия разрешимости уравнения (3), а также приведена оценка количества его решений. Решение уравнения осуществляется путем разбиения отношения R на «слои» $R_t, t \in T$, в каждом из которых уравнение имеет не более одного решения. В отдельном слое решение уравнения сводится к задаче перечисления множества входных вершин соответствующего слою графа $G_{R_t, b}$.

Теорема 2. Уравнение (3) имеет не более одного решения в каждом слое $R_t, t \in T$. Если граф $G_{R_t, b}$ не имеет циклов, то единственное решение уравнения состоит из всех атомов, соответствующих входным вершинам графа. Если $G_{R_t, b}$ имеет хотя бы один цикл, то уравнение решений не имеет.

На основе перечисленных результатов в следующем разделе будет показано, как аппарат уравнений может быть использован для формализации экспертных продукционных систем.

2. Экспертная продукционная система и ее представление LP-структурой

Чтобы описать применение методологии п.1 к исследованию баз знаний и моделированию обратного вывода, введем терминологию, связанную с экспертными продукционными системами. Рассматриваемый ниже пример основан на модели продукционных систем из [10], где описана их практическая реализация. Данная книга носит в большей мере учебный характер, однако ее идеи оказались удачными и в дальнейшем использовались и развивались в ряде работ по экспертным системам, в первую очередь – обучающим [13–14].

Итак, экспертные продукционные системы манипулируют множествами фактов и правил (продукций). *Факт* представляет единицу декларативной информации – некоторое суждение о внешнем мире.

Стандартным представлением факта является триплет вида «объект.атрибут = значение» [15] (например, «термометр.температура = высокая»). В [10] триплет редуцируется к паре «объект = значение»: например, «термометр.температура» и «термометр.изготовитель» просто считаются различными объектами. Среди множества объектов выделяются *объекты*

экспертизы, ради исследования которых разрабатывается экспертная система. Например, в базе знаний «Здоровье» [10] объектом экспертизы является «продолжительность жизни», которую пользователю может предсказывать экспертная система на основе значений некоторых «исходных объектов».

Во время своей работы продукционная система содержит так называемую *рабочую память*. Это некоторое подмножество фактов, которые на текущий момент работы считаются выполненными.

Правило состоит из предпосылки и заключения. *Предпосылка* обычно представляет собой выражение над фактами (например, их конъюнкцию). Предпосылка может быть выполненной (истинной) или невыполненной (ложной) при текущем состоянии рабочей памяти. Если предпосылка верна, то правило может быть применено. *Заключение* – это действие, которое можно осуществить, если верна предпосылка (например, добавить к рабочей памяти некоторые новые факты). *Применение правила* состоит в осуществлении действия заключения. Мы рассматриваем такие приложения, в которых заключение, как и предпосылка, является конъюнкцией фактов, и соответствующее заключению действие интерпретируется как «считать истинным». Таким образом, в нашем случае применение правила означает модификацию рабочей памяти – пополнение за счет фактов, справедливость которых вытекает из истинности выражения в заключении правила. Совокупность правил называется *базой знаний* (БЗ).

Прямой выводом в продукционной системе называется процесс циклического применения правил к содержимому рабочей памяти (его исходное состояние задано в начале работы), и соответственно получение в результате новых фактов, которые считаются истинными. Прямой вывод может осуществляться до тех пор, когда получение новых фактов станет невозможным, то есть при текущем содержимом рабочей памяти не окажется ни одной истинной предпосылки правила, заключение которого способно изменить рабочую память.

Противоположный процесс – *обратный вывод*. В нем по некоторому набору результирующих фактов (*гипотезе*), путем анализа правил в направлении от заключения к предпосылке, подтверждается или опровергается справедливость гипотезы при заданном исходном содержимом рабочей памяти. Это со-

держимое (*начальные факты*) экспертная система может получать из базы данных или путем опроса пользователя. При обратном выводе рабочая память также пополняется. Процесс решения поставленной задачи (определение значения заданного объекта экспертизы) будем также называть *экспертизой*.

Машиной вывода называется программа, осуществляющая прямой или обратный вывод в продукционной системе.

Предлагаемый подход к исследованию баз знаний и усовершенствованию обратного вывода основан на представлении наборов фактов и правил LP-структурой. Каждый элементарный факт изображается атомом решетки – булеана, предпосылка и заключение правила – соответствующими элементами решетки, а сами правила представляются парами бинарного отношения в LP-структуре. Нетрудно увидеть, что определяемое множеством правил бинарное отношение является рефлексивным и транзитивным. Его дистрибутивность также легко следует из возможности вывода совокупности фактов по частям. Согласно п.2, обратный продукционно-логический вывод в данной модели может быть осуществлен путем нахождения решений уравнения вида (1) с гипотезой в правой части. При этом искомым начальным прообраз гипотезы должен содержаться в рабочей памяти.

3. Релевантный обратный вывод

Покажем, какие возможности предоставляет описанный выше подход в сравнении с обычным обратным выводом.

Как известно, при исследовании алгоритмов принято анализировать их вычислительную сложность путем получения оценок относительно мощности множества исходных данных. Однако в случае продукционных систем (и логического вывода вообще) процесс нахождения подобных оценок обычно приводит к неудовлетворительным результатам. Действительно, имея множество N возможных элементарных фактов, можно сформировать до 2^N предпосылок и заключений потенциальных правил, что соответствующим образом скажется на оценках сложности алгоритмов машины вывода. Таким образом, важное значение приобретают альтернативные способы повышения эффективности системы.

В частности, существенную роль играет место хранения начальных фактов. В то время как

множество правил обычно может целиком располагаться в основной памяти компьютера, для начальных фактов такое хранение возможно далеко не всегда. Например, упомянутая выше БЗ «Здоровье» может применяться для тестирования различных пользователей, количество которых не ограничено, и каждый из них имеет собственную исходную информацию. В то же время единственная операция чтения с диска или, более того, получения информации от интерактивного пользователя, по времени выполнения способна перекрыть сотни и тысячи операций обращения к основной памяти. Таким образом, возникает идея организации такого процесса продукционно-логического вывода, при котором потребуется как можно меньше запросов к внешнему источнику информации.

При обычном обратном выводе [10] после выбора отправной гипотезы машина осуществляет просмотр содержащих гипотезу заключений правил, переходя к предпосылкам и в свою очередь рекурсивно проверяя их истинность. Когда в данном процессе в качестве очередной гипотезы встречается начальный факт (он не присутствует в правых частях правил), для проверки его истинности необходимо обращение к базе данных или пользователю. При этом не все проверяемые начальные факты в результате оказываются необходимыми для осуществления логического вывода, то есть порождения выбранной гипотезы.

Предлагаемый вывод на основе решения уравнений начинается с построения всех минимальных начальных прообразов в LP-структуре (п.2) для атомов, соответствующих значениям объекта экспертизы. Далее в построенном множестве достаточно найти тот прообраз, который содержит лишь истинные факты, после чего сразу можно сделать заключение о соответствующем значении объекта экспертизы. Самый простой путь в этом плане – просматривать прообразы последовательно, задавая пользователю вопросы о соответствующих начальных фактах (или обращаясь к базе данных за этими фактами). Этот способ уже дает определенные преимущества – исследуются лишь *минимальные* прообразы. Также предварительно можно исключить из процесса «противоречивые» прообразы, то есть содержащие одновременно альтернативные значения одного и того же объекта.

Существует более эффективный способ – приоритетный просмотр прообразов, содержа-

щих значения наиболее «релевантных» объектов. Таковыми в первую очередь могут считаться объекты, чьи значения присутствуют в максимальном количестве построенных прообразов. Тогда единственный отрицательный ответ пользователя на заданный вопрос исключает из рассмотрения сразу большое количество прообразов, что соответственно ускоряет исследование. Второй признак релевантности объекта – присутствие его значений в прообразах минимальной мощности. Таким образом, предпочтение отдается тем прообразам, проверка истинности которых потребует меньшего количества вопросов пользователю (или обращений к базе данных). Сочетая указанные два показателя релевантности, можно достичь результатов, по эффективности существенно превышающих возможности обычной машины вывода.

Поясним метод тривиальным примером, не претендующим на глубокий жизненный смысл. Пусть имеется следующая база знаний (из трех правил) для принятия решения о том, взять ли с собой зонтик при выходе на улицу.

- 1) **если** тучи **и** идти_пешком **и** выходишь_надолго **то** взять_зонтик;
- 2) **если** прогноз_плохой **и** идти_пешком **и** выходишь_надолго **то** взять_зонтик;
- 3) **если** идет_дождик **и** идти_пешком **то** взять_зонтик.

Для применения LP-вывода требуется вычислить все начальные прообразы гипотезы *взять_зонтик* (минимальные подмножества не выводимых из БЗ фактов, которые порождают факт *взять_зонтик*). В рассматриваемом примере это элементарно, поскольку они совпадают с предпосылками правил. Итак, гипотеза *взять_зонтик* имеет три начальных прообраза:

- 1) {тучи, идти_пешком, выходишь_надолго};
- 2) (прогноз_плохой, идти_пешком, выходишь_надолго);
- 3) {идет_дождик, идти_пешком}.

Далее среди них достаточно найти хотя бы один истинный, то есть состоящий целиком из выполненных фактов, тогда гипотеза окажется верной. Для тестирования каждого факта приходится обращаться к внешнему источнику информации, поэтому число таких проверок необходимо снижать. С этой целью, прежде чем выяснять истинность какого-либо факта, найдем наиболее релевантный из них.

Присвоив всем фактам нулевой приоритет, будем затем повышать его на 1, если факт со-

держится в максимальном (по сравнению с другими фактами) количестве прообразов. Поскольку факт *идти_пешком* встречается везде, его приоритет станет равным 1. Далее учитываем присутствие фактов в прообразах минимальной мощности (в данном случае – в прообразе №3). В результате повысятся приоритеты фактов *идет_дождик* (до 1) и *идти_пешком* (до 2). Таким образом, релевантным признается факт *идти_пешком*. Именно его истинность целесообразно проверить в первую очередь, обращаясь к внешнему источнику.

При отрицательном ответе на запрос о факте необходимо исключить из рассмотрения все содержащие его прообразы. В нашем случае это сразу решает «проблему зонтика» – все имеющиеся прообразы окажутся ложными («зонтик не брать!»). Если же ответ положителен, вычисляем следующий релевантный факт из еще непроверенных. Им окажется *идет_дождик*. Положительный ответ на запрос о данном факте также решает поставленную задачу – истинным будет прообраз №3 («зонтик брать!»), в противном случае процесс продолжается.

Как показывает рассмотренный пример, следование стратегии релевантности дает шансы решить задачу на ранних запросах, независимо от порядка вычисления прообразов. При этом стандартный обратный вывод [10] соответствовал бы некоторому линейному просмотру прообразов по мере их построения.

Переходя к терминологии LP-структур, сформулируем следующую задачу нахождения истинного прообраза. Даны конечная атомно-порожденная решетка $\mathbb{F}$ (атомы изображают элементарные факты) и на ней бинарное отношение R (представляет совокупность продукционных правил). Пусть выбран атом $b \in \mathbb{F}$, которому соответствует общее решение уравнения (3) – множество $\{X\}$ всех его минимальных начальных прообразов. Пусть также на множестве атомов решетки частично определена булева функция *True* (функция «истинности»), которая способна «доопределяться» путем обращения к внешнему источнику информации. В моделируемой продукционной системе интерпретация данной функции такова:

- $True(x)=1$, если соответствующий факт x содержится в рабочей памяти;
- $True(x)=0$, если достоверно известно, что x не может содержаться в рабочей памяти;

• $True(x) = null$, если проверка x еще не производилась.

Введем обозначения $T = \bigcup x_k (True(x_k) = 1)$; $F = \bigcup x_j (True(x_j) = 0)$. Необходимо найти такой элемент $X^0 \in \{X\}$, что $X^0 \subseteq T$ (если он существует). В процессе решения задачи требуется также обойтись как можно меньшим количеством «доопределений» функции $True$.

Для иллюстрации дадим упрощенные описания нерекурсивных алгоритмов обычного обратного вывода и релевантного LP-вывода.

// Обычный обратный вывод

$X^0 = null$

$X = getCurrPreImage(b)$

```
while  $X^0 = null$  and  $X \neq null$  do
  bool is_True = 1
  foreach  $x \subseteq X$  do
    if not  $x \subseteq T \cup F$  then is_True = Ask( $x$ )
    else is_True =  $x \subseteq T$ 
    if not is_True then break
  end
  if is_True then  $X^0 = X$ 
  else  $X = getCurrPreImage(b)$ 
```

end

Используемая здесь функция $Ask(x)$ спрашивает внешний источник об истинности атома x и в соответствии с ответом модифицирует множества T и F (то есть «доопределяет» функцию $True$). Функция $getCurrPreImage(b)$ находит очередной начальный прообраз для гипотезы b . Этот прообраз не обязательно минимален, поскольку не сравнивается с другими существующими начальными прообразами. Заметим также, что при обычном обратном выводе [10] проверка истинности каждого факта может производиться непосредственно перед его добавлением к формируемому прообразу, поэтому в представленном выше алгоритме не предусмотрены тесты вида $X \subseteq T$.

В следующем алгоритме, напротив, проверка на истинность подвергаются сразу целые прообразы.

// Релевантный LP-вывод

$X^0 = null$

$\{X\} = getPreImages(b)$

```
while  $X^0 = null$  and  $\{X\} \neq \emptyset$  do
   $k = getRelevantIndex(\{X\}, T)$ 
```

$Ask(x_k)$

```
foreach  $X_j \in \{X\}$  do
```

```
  if  $X_j \subseteq T$  then
```

```
     $X^0 = X_j$ 
```

```
    break
```

```
  end
```

```
  if  $X_j \cap F \neq \emptyset$  then  $\{X\} = \{X\} \setminus X_j$ 
```

```
end
```

end

В последнем алгоритме функция $getPreImages(b)$ решает уравнение (3) с правой частью b , то есть строит множество $\{X\}$ всех минимальных начальных прообразов для атома b .

Функция $getRelevantIndex(\{X\}, T)$ находит индекс k любого из наиболее релевантных и ранее не проверенных на истинность атомов, содержащихся в элементах текущего $\{X\}$. Она использует два упомянутых выше показателя релевантности начальных атомов с целью снижения количества вызовов функции $Ask(x_k)$. Один из возможных вариантов функции $getRelevantIndex$ представлен ниже.

```
int getRelevantIndex( $\{X\}, T$ )
```

// Инициализация приоритетов начальных атомов решетки

```
foreach  $x_k \in \mathbb{F}_0$  do Priority[k] = 0
```

// Нахождение минимальной мощности прообразов

```
int nMin = null;
```

```
foreach  $X_j \in \{X\}$  do
```

```
  if nMin = null or Length( $X_j$ ) < nMin
```

```
then nMin = Length( $X_j$ )
```

```
end
```

// Вычисление приоритетов начальных атомов

```
foreach  $x_k \in \mathbb{F}_0$  do
```

if not $x_k \subseteq T \cup F$ then // Рассматриваем лишь непроверенные атомы

```
  foreach  $X_j \in \{X\}$  do
```

```
    if  $x_k \subseteq X_j$  then
```

```
      Priority[k]++
```

```
      if Length( $X_j$ ) = nMin then
```

```
        Priority[k]++
```

```
      end
```

```
    end
```

```
  end
```

```
end
```

```

// Нахождение индекса наиболее реле-
вантного атома
int nRel = null;
foreach  $x_k \in \mathbb{F}_0$  do
    if nRel = null or Priority [k] > Priority
[nRel] then nRel = k
end
return nRel
end

```

При LP-выводе может производиться значи-тельно больше вычислений по сравнению с обычным обратным выводом, однако уменьша-ется количество вызовов функции $Ask(x)$ – обращений за информацией к внешнему источ-нику. Заметим, что формальное обоснование данного утверждения представляет собой ма-тематически недоопределенную задачу, суще-ственно зависящую от исходных данных. Для ее строгой постановки можно привлечь теорию вероятностей, однако этот путь, возможно, не даст реальных практических результатов. В п.5 настоящей статьи эффективность LP-вывода подтверждается приведенными результатами экспериментов – число внешних запросов в среднем снижается на 15–20%. Таким образом, на сегодня алгоритм LP-вывода можно назвать эвристическим в том смысле, что его преимущ-ества показаны лишь экспериментально.

Заметим, что метод релевантного обратного вывода, производя некоторую избыточную ра-боту, призван достичь общего выигрыша во времени. Данное обстоятельство имеет некото-рую аналогию с архитектурой современных процессоров, где с той же целью применяются технологии «спекулятивного» (избыточного) выполнения команд [16].

Вычисления показали, что при больших объемах БЗ процесс построения *всех* мини-мальных прообразов может потребовать чрез-мерного объема ресурсов компьютера. В связи с данным обстоятельством метод был модифи-цирован. Кластерно-релевантный LP-вывод предполагает последовательное построение кластеров начальных прообразов (подмножеств ограниченной мощности) с их динамическим релевантным исследованием. Модифицирован-ный LP-вывод дает достаточно эффективные (сопоставимые с LP-выводом) результаты для промышленных БЗ больших размеров. Приве-дем его обобщенный алгоритм.

```

// Кластерно-релевантный LP-вывод
 $X^0 = null$ 
while  $X^0 = null$  do
     $\{X\} = getPreImagesCluster(b)$ 
    while  $X^0 = null$  and  $\{X\} \neq \emptyset$  do
         $k = getRelevantIndex(\{X\}, T)$ 
         $Ask(x_k)$ 
        foreach  $X_j \in \{X\}$  do
            if  $X_j \subseteq T$  then
                 $X^0 = X_j$ 
                break
            end
            if  $X_j \cap F \neq \emptyset$  then  $\{X\} = \{X\} \setminus X_j$ 
        end
    end
end

```

Функция $getPreImagesCluster(b)$ выдает фиксированный (ограниченный по количеству элементов или времени вычислений) набор ми-нимальных начальных прообразов атома b , ко-торые на момент вызова функции не пересека-ются с множеством F .

На основе LP-структур и решения продук-ционно-логических уравнений может быть также организована верификация знаний. Во-первых, с помощью теоремы о логической ре-дукции продукционная БЗ может быть миними-зирована. Таким образом, в ней могут быть вы-явлены все избыточные правила. Во-вторых, БЗ можно исследовать на противоречивость. Сформулируем семантически противоречивую гипотезу, например, составленную из пары аль-тернативных значений объекта экспертизы. Решим далее продукционно-логическое урав-нение с данной гипотезой в качестве правой части. Если в результате будет найдено реше-ние, состоящее из непротиворечивых (совмес-тимых) фактов, можно сделать вывод о некор-ректности рассматриваемой базы знаний. Заметим, что пропозициональное исчисление не всегда способно идентифицировать проти-воречия, определяемые семантикой предметной области. Кроме того, общая математическая ло-гика часто имеет неэффективные реализации.

Преимущества LP-вывода демонстрируются в п.5 на результатах проведенных авторами экспериментов.

4. Применение к тестовым базам знаний

Авторами создан и сопровождается пакет программ LPExpert – интегрированная среда разработки продукционных экспертных систем [17]. Наряду с обычным обратным выводом программа содержит возможности решения продукционно-логических уравнений и реализации на этой основе релевантного вывода, а также верификации БЗ. Пакет LPExpert и файлы некоторых тестовых БЗ доступны для свободного скачивания с Internet-страницы <http://expert.vrn.ru/SD>.

В настоящем разделе с помощью тестовых примеров показываются преимущества обратного логического вывода на основе решения продукционно-логических уравнений, в сравнении со стандартным выводом.

В качестве первого демонстрационного примера рассмотрим БЗ «Здоровье» из книги [10]. Структура фактов и правил этой БЗ соответствует описанной выше в п.2. База знаний предназначена для выработки советов пациенту на основе оценки его состояния. Она выдает прогноз продолжительности жизни человека исходя из полученной от него информации о ряде факторов, влияющих на эту продолжительность. Данная БЗ служит лишь демонстрационным целям и не претендует на применение в медицине, хотя и содержит 77 правил. Мы используем ее «как есть», включая исходные недостатки, в русскоязычном переводе.

Итак, в LPExpert после загрузки файла БЗ «Здоровье» и запуска обычной экспертизы работает стандартная машина вывода. В целях экономии места сразу перечислим начальные факты, которые экспертная система формирует в результате диалога с пользователем, в хронологическом порядке ввода конкретных ответов:

возраст = 25–55

пол = м

вес = 55–85

сложение = мелкое

холестерин = норма

соль = норма

курение = нет

раса = европеоидная

характер = агрессивный

потребление_алкоголя = умеренное

Полученный результат экспертизы таков:

продолжительность = 62.

Исследуем результат проведенной экспертизы, построив с помощью LPExpert минимальные начальные прообразы факта *продолжительность* = 62. В результате решения уравнения с данной правой частью выдается множество минимальных прообразов общим количеством 49. Если все факты хотя бы одного из прообразов выполнены, то соответственно имеет место и факт *продолжительность* = 62. Путем просмотра минимальных прообразов можно убедиться в том, что ни один из них не совпадает с перечисленным выше множеством фактов, полученных от пользователя. Ближайший к этому множеству минимальный прообраз содержит на один факт меньше: в нем отсутствует значение *раса* = *европеоидная*. Это означает, что вычисленный при экспертизе на основе обычного обратного вывода прообраз факта *продолжительность* = 62 не является минимальным – указанный факт может быть выведен без использования значения *раса* = *европеоидная*. Следовательно, один из заданных пользователю вопросов оказался лишним. Избыточность вопросов говорит о нерациональной последовательности вывода. Отсюда возникает идея применения LP-вывода.

Как и выше, сразу перечислим факты, которые экспертная система формирует (на этот раз при LP-выводе) в порядке ввода пользователем ответов на вопросы (предполагается, что он не менял ответы на аналогичные вопросы):

вес = 55–85

сложение = мелкое

холестерин = норма

соль = норма

пол = м

возраст = 25–55

курение = нет

старт = да

Результат оказывается аналогичным предыдущей экспертизе:

продолжительность = 62.

На этот раз результат выведен, во-первых, из *минимального* прообраза, во-вторых, из прообраза минимальной мощности. Проведенный эксперимент показал, что применение LP-вывода способно существенно сократить количество обращений к пользователю – вместо 10 вопросов на этот раз было задано всего 8, с тем же результатом.

Еще одна возможность применения аппарата продукционно-логических уравнений состоит в

проверке базы знаний на противоречивость. С этой целью можно, например, пытаться строить минимальные начальные прообразы для пар значений объекта *продолжительность*. Наличие хотя бы одного непустого прообраза будет означать «возможность» одновременного вывода двух несовместимых значений, то есть противоречивость данной БЗ.

В диалоге просмотра значений БЗ будем пометать флажками одновременно по два значения объекта *продолжительность* и вычислять соответствующие им минимальные прообразы. После ряда безуспешных попыток окажется, что множество фактов $\{\text{продолжительность} = 48, \text{продолжительность} = 72\}$ имеет более 100 минимальных начальных прообразов. Вряд ли БЗ можно назвать корректной, если при одной и той же исходной информации она способна выдать два прогноза продолжительности жизни с разницей в 24 года!

В качестве еще одного приложения рассмотрим БЗ промышленного уровня «Электрики», разработанную в рамках исследований по интеллектуализации систем управления охраной объектов [18]. Она предназначена для получения всесторонних оценок знаний электромонтеров, обслуживающих пункт централизованной охраны. База знаний содержит 508 правил для получения оценок уровня знаний электротехники, электромонтажа, приборов охранной сигнализации и так далее. Полный текст данной БЗ с любезного согласия ее автора размещен на сайте <http://expert.vrn.ru/SD>.

Заметим, что на осмысленное использование БЗ столь значительного объема способен лишь специалист в достаточно узкой области. Экспертиза с использованием БЗ «Электрики» предполагает ввод пользователем порядка 70–100 ответов на вопросы, а результаты получаются в виде примерно двух десятков значений многозначного объекта экспертизы *знания*. В связи с данным обстоятельством, будем проводить экспертизу «наугад», давая на каждый вопрос системы первый из предлагаемых ответов и существенно не вдаваясь в смысл результатов. Количественные итоги стандартной экспертизы (на основе обычного обратного вывода) таковы: задано вопросов пользователю – 76, получено значений объекта *знания* – 13, включая оценку *основы_электротехники--два*.

Следующий шаг состоит в попытке провести для сравнения экспертизу на основе релевант-

ного LP-вывода. Однако данная попытка оказывается несостоятельной – после запуска LP-вывода программа «виснет». При более подробном изучении ситуации выясняется, что процесс решения соответствующих объекту экспертизы логических уравнений требует обработки свыше миллиарда слоев (см. п.1). Такой объем данных оказывается не под силу простому персональному компьютеру. Попробуем использовать для экспертизы кластеризованный LP-вывод. С этой целью в настройках программы зададим ограничение размера кластера, например, величиной в 200 прообразов. В результате новая экспертиза выполняется без видимых задержек, а ее количественные итоги оказываются следующими: задано вопросов пользователю – 57, получено значений объекта *знания* – 21, включая ту же оценку *основы_электротехники--два*.

Аналогичным образом проведем новые две соответствующие экспертизы (на основе обычного обратного, затем – кластерно-релевантного вывода), выбирая каждый последний возможный ответ вместо первого. Итоги стандартной экспертизы таковы: задано вопросов пользователю – 70, получено значений объекта *знания* – 12. Итоги кластеризованного LP-вывода: задано вопросов пользователю – 59, получено значений объекта *знания* – 21.

Анализируя результаты экспериментов с БЗ «Электрики», можно констатировать следующее:

- использование для экспертизы релевантного LP-вывода невозможно;
- при применении кластерно-релевантного LP-вывода снижение количества задаваемых пользователю вопросов составляет 15–25%.

С аналогичными целями авторы дополнительно провели порядка 100 автоматизированных тестов с «формальными» базами знаний, генерируемыми случайным образом. Результаты проведенных экспериментов так же свидетельствуют о преимуществах LP-вывода – количество задаваемых пользователю вопросов в среднем снижается на 15–20%.

Заключение

В работе представлены теоретические основы и показаны практические возможности применения теории LP-структур для ускорения обратного вывода, а также исследования и верификации баз знаний продукционного типа.

Предложены и эффективно реализованы алгоритмы усовершенствованного обратного вывода, направленные на минимизацию обращений к внешним источникам информации. Стратегия учета двух показателей релевантности (функция *getRelevantIndex*) снижает число внешних запросов в среднем на 15–20%. Возможно, существуют другие стратегии, которые позволят добиться более высоких результатов.

Представлен метод обнаружения противоречивости в базах знаний, основанный на решении продукционно-логических уравнений. Данный метод способен обнаруживать противоречия, которые происходят из семантики предметной области, то есть от конечного пользователя, что не всегда под силу общему пропозициональному исчислению.

Применение описанных методов продемонстрировано на простой архитектуре продукционных систем, чтобы сосредоточить внимание на самих методах. Аналогичный подход может быть применен и к продукционным системам более сложной структуры – многие модели в информатике имеют продукционный характер, а структуры представления информации часто являются иерархическими. Кроме того, представленная методология может быть применена к логическим системам более общего вида, в том числе и неклассическим.

В качестве возможной перспективы укажем также направление исследований по теоретическому обоснованию преимуществ алгоритмов релевантного LP-вывода на основе вероятностных методов.

Литература

1. Sowa J. F. Knowledge Representation: Logical, Philosophical and Computational Foundations / J. F. Sowa. – Brooks Cole Publishing Co., Pacific Grove, CA, 1999.
2. Бениаминов Е. М. Алгебраические методы в теории баз данных и представлении знаний / Е. М. Бениаминов. – М. : Научный мир, 2003. – 184 с.
3. Махортов С. Д. Логические отношения на решетках / С. Д. Махортов // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2003. – № 2. – С. 203–209.
4. Махортов С. Д. Логические уравнения на решетках / С. Д. Махортов // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2004, № 2. – С. 170–178.
5. Махортов С. Д. Релевантный обратный вывод и верификация логических программ на основе решения уравнений в LP-структурах / С. Д. Махортов // Методы и средства обработки информации: III Всероссийская научная конференция. Москва, 6–8 октября 2009 г.: Сб. трудов / Под ред. Л.Н. Королева. – М. : ВМК МГУ, 2009. – С. 143–148.
6. Forgy, C. L. Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence, 19(1):17-37 (1982).
7. Miranker, D: Performance estimates for the DADO machine: A comparison of TREAT and Rete. In Fifth Generation Computer Systems, ICOT, Tokyo, (1984).
8. Miranker, D., Brant, D., Lofaso, B. and Gadbois, D.: On the Performance of Lazy Matching in Production Systems. Proc. National Conference on Artificial Intelligence (1990).
9. Lee, Y. H., Yoo, S.I.: Integration of Forward and Backward Inferences Using Extended Rete Networks. In: Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 9th, Proceedings of the Ninth International Conference (1997) 339-345.
10. Sowyer, B., Foster, D.: Programming Expert Systems in Pascal. John Wiley, Inc. (1986).
11. Aho, A.V., Garey, M.R., Ulman, J.D.: The transitive reduction of a directed graph. SIAM J. Computing 1:2 (1972) 131-137.
12. Birkhoff, G.: Lattice Theory. Providence, Rhode Island (1967).
13. Giraud-Carrier, C.G., Martinez, T.R.: An Integrated Framework for Learning and Reasoning. Journal of Artificial Intelligence Research v.3, (1995) 147-185.
14. Zimmermann, T., Bomme, P.: Toward intelligent object-oriented scientific applications. In: Engineering Computational Technology, B. H. Topping and Z. Bittnar, Eds. Civil-Comp press, Edinburgh, UK (2002) 271-311.
15. Doorenbos, R. B. Production Matching for Large Learning Systems. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-22942., Carnegie Mellon University (1995).
16. Таненбаум Э. Архитектура компьютера : пер. с англ. / Э. Таненбаум. – СПб. : Питер, 2002. – 704с.
17. Махортов С. Д. Интегрированная среда логического программирования LPExpert / С. Д. Махортов // Информационные технологии. – 2009. – № 12. – С. 65–66.
18. Засыпкин А.В. Экспертная система поддержки решений по рациональной организации охранно-пожарной сигнализации : автореф. дис. ... канд. техн. наук: 05.13.11 / А. В. Засыпкин. – М. : МИП, 1993. – 20 с.

Болотова Светлана Юрьевна. Преподаватель Воронежского госуниверситета (ВГУ). Окончила ВГУ в 2009 г. Научные интересы: логический вывод, искусственный интеллект. E-mail: cbeta555@mail.ru.

Махортов Сергей Дмитриевич. Заведующий кафедрой Воронежского госуниверситета (ВГУ). Окончил ВГУ в 1980 г. Доктор физ.-мат. наук, доцент. Количество печатных работ – 92, из них монографий – 4. Научные интересы: алгебраические системы, искусственный интеллект, теория программирования. E-mail: sd@expert.vrn.ru.