

Комплексная верификация продукционных баз знаний с использованием V^{TF} -логик

Л.В. Аршинский, А.А. Ермаков, М.С. Нитежук

Иркутский государственный университет путей сообщения, Иркутск, Россия

Аннотация

Рассматривается комплексная процедура верификации продукционных баз знаний с использованием логик с векторной семантикой в варианте V^{TF} -логик при специальном представлении фактов и правил. Описанная техника позволяет решать такие задачи верификации, как выявление несвязанных фактов, выявление незавершённых продукций, выявление логических кругов, контроль соответствия между множеством гипотез и множеством терминальных фактов, выявление противоречий, выявление молчащих продукций, выявление нештатных обрывов цепочек вывода. Продукции в базе знаний упорядочиваются причинно-следственным образом так, что если один и тот же факт входит в правую часть одной продукция и левую часть другой, первая продукция всегда выполняется раньше. В результате процедура верификации имеет линейную сложность по числу правил и экспоненциальную по числу стартовых фактов. Объём вычислений можно существенно уменьшить, выделяя группы фактов, относящихся к конкретной гипотезе. Новым является применение для верификации аппарата логик с векторной семантикой, которые сохраняют способность к выводу при аномальных значениях истинности. Это позволяет, в частности, использовать машину вывода для динамической верификации знаний. В результате не требуется вводить в систему дополнительные архитектурные элементы (например, таблицы решений), создавать внешние верифицирующие программы и т.п. Получение решения обеспечивается штатными средствами экспертной системы. Статическая верификация обеспечивается специальным представлением фактов и правил.

Ключевые слова: экспертные системы, продукционная модель знаний, верификация, логики с векторной семантикой.

Цитирование: Аршинский, Л.В. Комплексная верификация продукционных баз знаний с использованием V^{TF} -логик / Л.В. Аршинский, А.А. Ермаков, М.С. Нитежук // Онтология проектирования. – 2020. – Т. 10, №1. – С.112-120. – DOI: 10.18287/2223-9537-2020-10-1-112-120.

Введение

Проблема верификации баз знаний (БЗ) экспертных систем (ЭС) продукционного типа известна давно. В [1] отмечается, что некоторые из первых результатов в этом направлении были представлены в [2]. В [3, 4] была подчеркнута важность этой проблемы. Уже на ранних этапах сложился терминологический аппарат, и были сформулированы такие основные требования к качеству БЗ, как полнота (способность работать во всех допустимых входных ситуациях) и отсутствие противоречий в ходе вывода [5-7]. В [7] рассмотрен ряд известных на тот момент систем и методов верификации БЗ и отмечено, что нет универсального подхода к верификации, и полноценной она становится, когда эти методы применяются в комплексе или для специальных БЗ. Сходное мнение выражено в [8]. Это делает, в частности, актуальными работы по верификации БЗ для отдельных предметных областей (см. напр. [9]). Для обнаружения и удаления артефактов было предложено генерировать соответствующие метазнания, тесты [7, 10], использовать подходы, аналогичные доказательству правильности компьютерных программ, разработаны языки спецификаций [11-13].

В частности, в работе [14] рассмотрены вопросы применения логик с векторной семантикой для обнаружения дефектов продукционных БЗ. Основное внимание было уделено артефактам, ведущим к появлению противоречий, обрыву цепочек вывода. Показано, что при использовании машины вывода, работающей на основе логик с векторной семантикой в варианте V^{TF} -логик, в которых истинность формализуется вектором $\langle a^+; a^- \rangle$ (где $a^+, a^- \in [0,1]$; a^+ – степень уверенности, что суждение a истинно, а a^- , – что оно ложно), противоречивость, неопределённость или ложность посылок в общем случае не обрывает вывод, но порождает проходящие через всю последующую цепочку аномальные значения истинности, которые можно отследить. Это позволяет предложить метод поиска артефактов: организация прямого вывода для различных допустимых сочетаний истинности входных фактов с последующей регистрацией аномального результата. Если допустимый вход порождает «неправильный» выход, это считается следствием артефакта. Далее организуется обратная трассировка вывода для обнаружения причин аномалии.

Одной из проблем верификации является экспоненциальный рост вычислений с увеличением объёма БЗ. Верификация во многом реализуется с помощью дополнительных компонентов, выходящих за рамки «штатных» средств ЭС. В предлагаемой работе описан комплекс процедур и алгоритмов, развивающих подход к проблеме верификации, представленный в [14]. При этом верификация может выполняться без включения в состав архитектуры ЭС дополнительных составляющих.

1 Подготовка и статическая верификация БЗ

Подготовка БЗ начинается с формирования множества атомарных утверждений (базы фактов). Каждое утверждение представляется четвёркой:

$$(1) \quad A = (a, C_1, C_2, \|a\|).$$

Здесь a – содержательная часть утверждения, например, «Температура воздуха высокая» или «Скорость объекта больше 30 м/сек», «У пациента наблюдается сыпь» и т.п.; C_1 и C_2 – счётчики числа вхождений A , соответственно, в левую и правую часть правил;

$$(2) \quad \|a\| = \langle [a^+_1; a^+_2]; [a^-_1; a^-_2] \rangle,$$

– вектор истинности a в интервальном представлении. Если a строго истинно, $\|a\| = \langle [1; 1]; [0; 0] \rangle$; если строго ложно – $\langle [0; 0]; [1; 1] \rangle$. Когда свидетельства «за» или «против» a отсутствуют (т.е. имеет место неопределённость), $\|a\| = \langle [0; 0]; [0; 0] \rangle$. Наконец для полностью противоречивого случая $\|a\| = \langle [1; 1]; [1; 1] \rangle$. Первоначальное значение истинности для всех фактов выставляется в неопределённость. Далее она задаётся пользователем или вычисляется в ходе присоединённого вывода.

В конкретной реализации (1) может дополняться чем-либо ещё, что необходимо с точки зрения разработчика, но эти четыре составляющих рассматриваются как необходимые.

Каждое вхождение факта A в ту или иную продукцию сопровождается увеличением соответствующего счётчика C_1 или C_2 на единицу, а его удаление – уменьшением. Это позволяет разбить множество фактов на подмножества:

- стартовые факты ($C_1 > 0, C_2 = 0$);
- промежуточные факты ($C_1 > 0, C_2 > 0$);
- терминальные факты (гипотезы) ($C_1 = 0, C_2 > 0$);
- неиспользуемые факты ($C_1 = 0, C_2 = 0$).

Представление (1) решает проблему выявления неиспользуемых фактов, а также позволяет разработчикам сопоставить фактический набор гипотез с тем, что был запланирован

изначально. Это, в частности, снимает один из вопросов обнаружения недостижимых гипотез [1].

Далее факты связываются в правила R , которые представляются пятёркой:

$$(3) \quad R = (i, K_1, K_2, \|i\|, L).$$

Здесь i – импликация (ядро продукции), K_1 и K_2 – счётчики числа фактов в левой и правой частях правила, $\|i\|$ – истинность ядра, также представляющая собой вектор с интервальными значениями:

$$\|i\| = \langle [i^+_1; i^+_2]; [\bar{i}^-_1; \bar{i}^-_2] \rangle.$$

Значение $\|i\|$ задаётся разработчиками. L – положение (уровень) продукции в причинно-следственном порядке.

Счётчики K_1 и K_2 в (3) позволяют автоматизировать выявление незавершённых правил, в которых отсутствует антецедент или консеквент (также один из возможных артефактов [1]).

Серьёзной проблемой разработки ЭС является рост объёма вычислений с увеличением БЗ, если не пользоваться специальными методами и алгоритмами. Классическим примером такого метода служит алгоритм Rete [15]. Для решения задачи верификации на завершающем этапе формирования БЗ предлагается использовать причинно-следственное упорядочение продукции. Под причинно-следственным упорядочением понимается такая расстановка правил, что если две продукции $R1$ и $R2$ содержат общее утверждение a , причём в одном правиле, например $R1$, оно входит в консеквент, а в другом ($R2$) в антецедент, правило $R1$ при выводе всегда предшествует $R2$. В результате для любой продукции R все логически предшествующие ей продукции выполняются прежде R [16]. Логический вывод при этом выполняется за один проход БЗ, а объём вычислений с увеличением числа правил растёт не экспоненциально, а линейно. Если использовать машину вывода, в том числе для задач верификации [14], это существенно упрощает дело.

Подобное упорядочение при неизменной БЗ выполняется однократно и до начала работы с системой (редактирование БЗ требует переупорядочения).

Упорядочение может выполняться по следующему принципу:

- начальные значения уровней всех правил, антецедент которых состоит только из стартовых фактов, выставляется в 0 (т.е. $L = 0$);
- если в антецедент правила входят факты, входящие в консеквенты правил с уровнями L_1, \dots, L_n , текущему правилу приписывается уровень $L = \max(L_1, \dots, L_n) + 1$;
- процедура выполняется за несколько проходов БЗ, пока не прекратится изменение уровней.

При следовании такому принципу правила одного уровня никогда не ссылаются друг на друга, а правила уровня L никогда не ссылаются на правила с уровнями, предшествующими L . Обеспечивается однопроходность вывода и линейный рост объёма вычислений с увеличением БЗ. Часть вывода, требующая больших затрат времени, переносится на этап настройки БЗ и не затрагивает пользователя ЭС.

Процедура может войти в бесконечный цикл, если в БЗ имеются причинно-следственные петли (выход правила попадает на вход одного из предшествующих правил соответствующей цепочки вывода), что требует ограничения на число возможных уровней, которое при отсутствии петель равно длине максимальной цепочки. Если задать достаточно большое ограничивающее число, его достижение для какой-то группы продукции сигнализирует о наличии причинно-следственной петли в этой группе – потенциальном логическом круге. Это позволяет заранее выявлять подобные аномалии.

Таким образом, на этапе формирования БЗ контролируются такие статические аномалии, как несвязанные факты (включая гипотезы), незавершённые продукции, логические круги,

выявляется принадлежность гипотез множеству терминальных фактов (т.е. действительно ли они гипотезы для машины вывода). Дополнительный контроль соответствия между терминальными фактами и гипотезами обеспечивается также причинно-следственным упорядочением правил.

2 Динамическая верификация БЗ

Под динамической верификацией понимается выявление артефактов, способных проявиться только при работе машины вывода. К таким артефактам относятся получение противоречивого заключения, незапланированный обрыв цепочки вывода, «молчание» (никогда не запускающиеся) правила.

В [14] эти ситуации обсуждались с точки зрения их выявления средствами V^{TF} -логик. Основная идея сводится к тому, что ситуации, порождающие эти артефакты, как то:

- совместное получение утверждения и отрицания какого-либо факта (противоречие),
- постоянно ложный антецедент или (в векторном случае) постоянно неопределённый антецедент (все компоненты вектора истинности близки к нулю) - не препятствуют процедуре логического вывода, но порождают заключения с аномальными значениями истинности соответствующего типа. Этот факт устанавливается на завершающем этапе вывода выявлением гипотез с такими же аномальными значениями истинности. Обратной трассировкой цепочки вывода можно установить причину аномалии. Важной составляющей такого вывода служит процедура объединения свидетельств, которая в V^{TF} -случае опирается на одну из следующих операций [16].

- «10»-композицию, которой называется суждение (первая форма дизъюнкции), образованное из a и b с вектором истинности:

$$\|a \vee b\| = \langle a^+ \oplus b^+; a^- \bullet b^- \rangle.$$

- «01»-композицию с вектором истинности (первая форма конъюнкции):

$$\|a \& b\| = \langle a^+ \bullet b^+; a^- \oplus b^- \rangle.$$

- «11»-композицию с вектором истинности (вторая форма дизъюнкции):

$$\|a \vee_2 b\| = \langle a^+ \oplus b^+; a^- \oplus b^- \rangle.$$

- «00»-композицию с вектором истинности (вторая форма конъюнкции):

$$\|a \&_2 b\| = \langle a^+ \bullet b^+; a^- \bullet b^- \rangle.$$

Первые формы дизъюнкции и конъюнкции аналогичны дизъюнкции и конъюнкции в классическом и нечётком случаях, вторые формы – специфика V^{TF} -логик. Здесь \bullet и \oplus – триангулированная (треугольные – *triangle*) норма и ко-норма в инфиксной записи с дополнительным свойством [17]:

$$x \bullet y = 1 - (1 - x) \oplus (1 - y);$$

либо:

$$x \oplus y = 1 - (1 - x) \bullet (1 - y).$$

Это свойство связывает векторное и нечёткое представления истинности в случае, когда $a^- + a^+ = 1$ [18].

В [14] перечислены возможные стратегии объединения свидетельств, часть которых способствует выявлению артефактов. С точки зрения выявления противоречий подходят следующие две.

1) объединение по схеме «10»/«11»-композиции – стратегия *нарастающего доверия*. Свидетельства в пользу истинности b , как и свидетельства в пользу истинности $\neg b$, объединя-

ются отдельно по правилу «10»-композиции (дизъюнкции). Полученные после этого векторы $\|b\|$ и $\|\neg b\|$ «складываются» в единый вектор по правилу «11»-композиции. При таком объединении доверие к b возрастает по мере получения утвердительных доказательств факта, доверие к $\neg b$ – по мере вывода его отрицаний. Итоговый вектор объединяет свидетельства как за, так и против.

- 2) объединение по схеме «11»-композиции – стратегия *полного доверия*. Доверяем и подтверждающим и опровергающим свидетельствам, это упрощённый вариант стратегии нарастающего доверия. Здесь быстрее растёт степень противоречивости промежуточных фактов и гипотез, что проще с алгоритмической точки зрения.

Оба подхода при получении взаимоисключающих утверждений b и $\neg b$ генерируют вектор истинности, фиксирующий противоречие. Т.е., если по некоторой ветви вывода получено заключение b с истинностью $\|b\| = \langle 1; 0 \rangle$, а по другой $\neg b$ также с истинностью $\|\neg b\| = \langle 1; 0 \rangle$, то объединение на основе «11»-композиции даёт (с учетом, что $\|\neg b\| = \langle b^-; b^+ \rangle$):

$$\|b\| = \langle 1 \oplus 0; 0 \oplus 1 \rangle = \langle 1; 1 \rangle - \text{полное противоречие.}$$

Возможна ситуация, когда противоречие имеет неявную форму: из одних и тех же посылок генерируются логически несовместимые заключения b и c [1, 19]. Введение дополнительной продукции с ядром $b \rightarrow \neg c$ (или $c \rightarrow \neg b$) позволит выявить и этот случай.

Верификация выполняется в ходе прямого вывода и предполагает перебор всех допустимых значений истинности входных фактов. Поскольку предлагаемая модель истинности непрерывна, следует ограничиться конечным набором проверочных комбинаций. Чтобы избежать маскирующего влияния нестрогих значений истинности, истинность малых посылок целесообразно задавать в виде допустимых комбинаций строгой истины $\langle [1; 1]; [0; 0] \rangle$ и/или строгой лжи $\langle [0; 0]; [1; 1] \rangle$ при строгой истинности импликации.

В результате на этом этапе возможно выявление таких артефактов, как противоречие и неустранимая ложность или неопределённость antecedентов продукций [14]. Последнее в классическом случае приводит к обязательному обрыву цепочки вывода, а в рассматриваемой технике порождает аномальную истинность гипотезы, сигнализирующую об артефакте.

Если дополнить (3) счётчиком числа обращений к продукции, легко установить «молчащие» продукции, к которым не было ни одного обращения. Однако при использовании причинно-следственного порядка этот артефакт можно обнаружить уже на этапе статического анализа.

3 Снижение вычислительной сложности

Остаётся прояснить вопрос о процедуре выставления значений истинности стартовых фактов – посылок. В предельном случае N посылок порождают 2^N комбинаций истинности. Даже для небольшой БЗ, содержащей несколько десятков входных фактов, это приводит к взрывообразному росту числа проверок. Простой перебор может оказаться невозможным. Количество проверок необходимо уменьшить без потери полноты перебора. Для решения этой задачи предлагается использовать то обстоятельство, что в основе каждой гипотезы лежит относительно небольшое количество стартовых посылок. Тогда для каждой гипотезы можно выделять только относящиеся к ней факты. Поскольку в реальных задачах обычно n существенно меньше N , где n – число стартовых фактов гипотезы, получаем кардинальное снижение вычислительной нагрузки. Далее приводится дополненная схема алгоритма, представленного в [14].

- 1) объявить все продукции строго истинными (запомнив, при необходимости, значения истинности, выставленные экспертами).

- 2) выделить терминальный факт (гипотезу).
- 3) установить все связанные с ним стартовые факты.
- 4) выставить значения истинности стартовых фактов в одну из допустимых комбинаций - $\langle 1; 0 \rangle$ или $\langle 0; 1 \rangle$, - а истинность промежуточных и терминальных - в $\langle 0; 0 \rangle$.
- 5) выполнить прямой логический вывод, получив истинность гипотезы.
- 6) проверить истинность заключения на допустимость.
- 7) если заключение недопустимо:
 - 7.1) выполнить обратную трассировку вывода для выяснения источника проблемы – артефакта БЗ;
 - 7.2) устранить артефакт (или внести его в соответствующий список);
 - 7.3) вернуться к шагу 4.
- 8) если не все возможные входные значения посылок проверены, вернуться к шагу 4.
- 9) если не все возможные гипотезы проверены, вернуться к шагу 2.
- 10) вернуть исходные значения истинности продукций.
- 11) стоп.

Такая верификация не требует генерирования дополнительных структур данных, например таблиц принятия решений или их аналогов.

Заключение

В работе представлена комплексная процедура верификации продукционных БЗ на основе V^{TF} -логик с введением в факты и правила дополнительных элементов, а также специальным упорядочением правил. Решаются следующие вопросы:

- выявление несвязанных фактов;
- выявление незавершённых продукций;
- выявление логических кругов;
- контроль соответствия между множеством гипотез и множеством терминальных фактов;
- выявление противоречий;
- выявление «молчащих» продукций;
- выявление нештатных обрывов цепочек вывода.

Верификация имеет линейную сложность по числу правил и экспоненциальную по числу стартовых фактов. Объём вычислений можно существенно уменьшить, выделяя факты, относящиеся к конкретной гипотезе.

Представленная процедура не исчерпывает всех проблем верификации (например, дублирования правил) и применима к заранее спроектированным под неё БЗ. Однако при этом не требуется введение в состав ЭС дополнительных архитектурных элементов в виде таблиц решений и т.п. Полученные решения задачи обеспечиваются штатными средствами ЭС.

Список источников

- [1] *Nguyen, T.A.* Knowledge Base Verification / T.A. Nguyen, W.A. Perkins, J.T. Laffey, D. Pecora // AI Magazine. - 1987. - V. 8, № 2. – P. 69-75.
- [2] *Davis, R.* Applications of Meta-Level Knowledge to the Construction, Maintenance, and Use of Large Knowledge Bases / R. Davis // Ph D. diss, Dept of Computer Science, Stanford Univ., 1976.
- [3] *O'Leary, D.* Design, development and validation of expert systems: A survey of developers / D. O'Leary // In M. Ayel and J-P. Laurent (Eds.) Validation, Verification and Test of Knowledge-Based Systems. 1991. – P. 3-20.
- [4] *Hamilton, D.* State-of-the-practice in knowledge-based system verification and validation / D. Hamilton, K. Kelley, C. Culbert // Expert Systems with Applications. - 1991. – 3. - P. 403–410.
- [5] *Benbasat, I.* A framework for the validation of knowledge acquisition / I. Benbasat, J.S. Dhaliwal // Knowledge Acquisition. - 1989. – 1. - P. 215–233.

- [6] **Laurent, J.-P.** Proposals for a valid terminology in KBS validation / J.-P. Laurent // In B. Neuman (Ed.) Proceedings of the 10th European Conference on Artificial Intelligence (ECAI-92), 1992. - P. 829–834.
- [7] **Preece, A.D.** Evaluation of Verification Tools for Knowledge-Based Systems / A.D. Preece, S. Talbot, L. Vignollet // Int. J. Hum.-Comput. Stud. - 1997. - V. 47. – P. 629-658.
- [8] **Логунова, Е.А.** Обзор подходов к разрешению недостатков продукционной базы знаний системы логического вывода / Е.А. Логунова // Современные наукоемкие технологии. - 2015. - № 9. – С. 46-48.
- [9] **Racunas, S.A.** A case study in pathway knowledge base verification / S.A. Racunas, N.H. Shan, N.V. Fedoroff // BMC Bioinformatic. – 2006. 7: 196. (Published online 2006 Apr 8). - DOI: 10.1186/1471-2105-7-196.
- [10] **Bindilatti, A. de A.** Verification and validation of knowledge bases using test cases generated by restriction rules / A. de A. Bindilatti, A.E.A da Silva // International Journal of Artificial Intelligence and Expert Systems. - 2012. - V. 3, Issue 4. – P. 117-125.
- [11] **Antonioni, G.** Verification and Validation of Knowledge-Based Systems. Report on Two 1997 Events / G. Antonioni, F. van Harmelen, R. Plant, J. Vanthienen // AI Magazine: Workshop Report. - 1998. - V.19, № 3. – P. 123-126.
- [12] **Бобина, В.А.** Верификация и подтверждение правильности (V&V) системы управления базой знаний с помощью формальных спецификаций / В.А. Бобина // Международный научный журнал «Символ науки». - 2015. - №5. – С.17-20.
- [13] **Терновой, М.Ю.** Формальная спецификация свойств баз нечетких знаний Мамдани на основе метаграфа / М.Ю. Терновой, Е.С. Штогрин // Вісник Харківського національного університету імені В.Н. Каразіна. Серія «Мат. моделювання. Інформаційні технології. Автоматизовані системи управління», 2015. - Вып. 27. – С. 157-171.
- [14] **Аршинский, Л.В.** Логика с векторной семантикой как средство верификации баз знаний / Л.В. Аршинский, А.А. Ермаков, М.С. Нитежук // Онтология проектирования. - 2019. - Т.9, № 4(34). – С. 510-521. - DOI: 10.18287/2223-9537-2019-9-4-510-521.
- [15] **Forgy, Ch.L.** Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem / Ch.L. Forgy // Artificial Intelligence. - 1982. - V. 19, № 1. - P 17-37.
- [16] **Аршинский, Л.В.** Особенности работы машины вывода системы моделирования правдоподобных рассуждений «Гераклит» / Л.В. Аршинский // Информационные и математические технологии в науке и управлении. - 2016. - №2. – С.18-29.
- [17] **Gottwald, S.** Treatise on Many-Valued Logics / S. Gottwald. - Leipzig, 2000. - 604 p.
- [18] **Аршинский, Л.В.** Содержательный и формальный выводы в логиках с векторной семантикой / Л.В. Аршинский // Автоматика и телемеханика. - 2007. - № 1. - С. 153-162.
- [19] **Проскуряков, Д.П.** Управление разрешением конфликтов в продукционных экспертных системах / Д.П. Проскуряков // Вестник Иркутского государственного технического университета. - 2015. - № 8. - С. 47-51.

Сведения об авторах

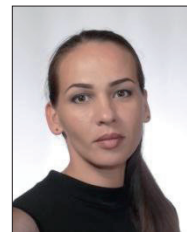


Аршинский Леонид Вадимович, 1957 г. рождения. Окончил Иркутский государственный университет в 1979 г., д.т.н. (2008). Заведующий кафедрой «Информационные системы и защита информации» Иркутского государственного университета путей сообщения. В списке научных трудов более 200 работ в области околоэкранный аэродинамики, распознавания образов, моделирования правдоподобных рассуждений, агрегированного оценивания систем и др. AuthorID (РИНЦ): 520252; Author ID (Scopus): 57193195356; Researcher ID (WoS): C-3869-2013; ORCID 0000-0001-5135-7921. larsh@mail.ru.

Ермаков Анатолий Анатольевич, 1951 г. рождения. Окончил Киевское высшее военное авиационное инженерное училище в 1976 г., к.т.н. (1982). Профессор кафедры «Информационные системы и защита информации» Иркутского государственного университета путей сообщения. В списке научных трудов около 70 работ в области технической диагностики, надежности и эксплуатации сложных систем, распознавания образов и др. AuthorID (РИНЦ): 861476; Researcher ID (WoS): AАН-4024-2020; ORCID 0000-0002-9312-9916. anatolij.ermackow2011@yandex.ru.



Нитежук Марина Сергеевна, 1984 г. рождения. Окончила Иркутский государственный университет путей сообщения в 2010 г. Старший преподаватель и аспирант кафедры «Информационные системы и защита информации» Иркутского государственного университета путей сообщения. В списке научных трудов 9 работ в области информационных систем и технологий. AuthorID (РИНЦ): 1007958; Author ID (Scopus): 57214999113; Researcher ID (WoS): AАН-4012-2020; ORCID 0000-0003-3420-8312. *marino_@mail.ru*.



Поступила в редакцию 10.02.2020, после рецензирования 10.03.2020. Принята к публикации 15.03.2020.

Complex verification of rule-based knowledge bases using V^{TF} -logic

L.V. Arshinskiy, A.A. Ermakov, M.S. Nitezjuk

Irkutsk State Transport University, Irkutsk, Russia

Abstract

In this paper, we consider a complex procedure for verifying rule-based knowledge bases using logics with vector semantics in the form of V^{TF} -logic with a special representation of facts and rules. The described technique allows solving verification tasks such as identifying unrelated facts, identifying incomplete rules, identifying logical circles, checking the correspondence between a set of hypotheses and a set of terminal facts, identifying contradictions, identifying silent rules, and identifying abnormal breaks in the inference chains. Rules in the knowledge base are arranged in a causal way. It means if the same fact is included in the right part of one rule and the left part of another, the first rule is always performed earlier. As a result, the verification procedure is linear in terms of the number of rules and exponential in terms of the number of starting facts. However, the amount of calculations can be significantly reduced by selecting groups of facts related to a particular hypothesis. The novelty of the study is the use of logics with vector semantics for rules verification. This retains the ability to inference in case of anomalous values of truth. This logic provides to use the inference engine for dynamic verification of knowledge. As a result, you do not need to include additional architectural elements into the system (for example, rule set dependency chart), create external verification programs, etc. In fact everything is provided by regular means of the expert system. Static verification is provided by a special representation of facts and rules.

Keywords: *expert systems, rule-based systems, verification, logic with vector semantics*

Citation: *Arshinskiy LV, Ermakov AA, Nitezjuk MS. Complex verification of rule-based knowledge bases using V^{TF} -logic [In Russian]. *Ontology of designing*. 2020; 10(1): 112-120. DOI: 10.18287/2223-9537-2020-10-1-112-120.*

References

- [1] *Nguyen TA, Perkins WA, Laffey JT, Pecora D.* Knowledge Base Verification. *AI Magazine*; 1987; 8(2): 69-75.
- [2] *Davis R.* Applications of Meta-Level Knowledge to the Construction, Maintenance, and Use of Large Knowledge Bases. Ph D. diss, Dept of Computer Science, Stanford Univ.; 1976.
- [3] *O'Leary D.* Design, development and validation of expert systems: A survey of developers. In M. Ayel & J-P. Laurent, Eds. *Validation, Verification and Test of Knowledge-Based Systems*; 1991: 3-20.
- [4] *Hamilton D, Kelley K, Culbert C.* State-of-the-practice in knowledge-based system verification and validation. *Expert Systems with Applications*; 1991; 3: 403-410.
- [5] *Benbasat I, Dhaliwal JS.* A framework for the validation of knowledge acquisition. *Knowledge Acquisition*; 1989; 1: 215-233.
- [6] *Laurent J-P.* Proposals for a valid terminology in KBS validation. In B. Neuman, Ed. *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI-92)*; 1992; 829-834.
- [7] *Preece AD, Talbot S, Vignollet L.* Evaluation of Verification Tools for Knowledge-Based Systems. *Int. J. Hum.-Comput. Stud*; 1997; 47: 629-658.
- [8] *Logunova EA* Approaches to resolving the deficiencies of production knowledge base of inference system [in Russian]. *Modern High Technologies*; 2015; 9: 46-48.

- [9] **Racunas SA, Shan NH, Fedoroff NV.** A case study in pathway knowledge base verification. *BMC Bioinformatic*; 2006, 7: 196. DOI: 10.1186/1471-2105-7-196.
 - [10] **Bindilatti AA, da Silva AEA.** Verification and validation of knowledge bases using test cases generated by restriction rules. *International Journal of Artificial Intelligence and Expert Systems (IJAE)*; 2012; 3(4): 117-125.
 - [11] **Antonioni G, van Harmelen F, Plant R, Vanthienen J.** Verification and Validation of Knowledge-Based Systems. Report on Two 1997 Events. *AI Magazine: Workshop Report*, 1998; 19(3): 123-126.
 - [12] **Bobina VA.** Verification and validation (V&V) of the knowledge base management system using formal specifications [in Russian]. *International Scientific Journal "Symbol of Science"*; 2015; 5: 17-20.
 - [13] **Ternovoy MYu, Shtogrina ES.** Formal specification of properties of Mamdani fuzzy knowledge bases based on metagraph [in Russian]. *Kharkov University bulletin*; 2015; 27: 157-171.
 - [14] **Arshinskiy LV, Ermakov AA, Nitezuk MS.** Logic with vector semantic as a means of knowledge bases verification [In Russian]. *Ontology of Designing*; 2019; 9(4): 510-521. DOI: 10.18287/2223-9537-2019-9-4-510-521.
 - [15] **Forgy ChL.** Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem. *Artificial Intelligence*; 1982; 19(1): 17-37.
 - [16] **Arshinskiy LV** Features of the inference engine of the plausible reasoning modeling system "Heraclitus" [in Russian]. *Information and Mathematical Technologies in Science and Management*; 2016; 2: 18-29.
 - [17] **Gottwald S.** Treatise on Many-Valued Logics. Research Studies Press; 2001.
 - [18] **Arshinskii LV.** Substantial and formal deductions in logics with vector semantics. *Automation and remote control*; 2007; 68(1): 139-148.
 - [19] **Proskuriakov DP.** Conflict resolution management in rule-based expert systems [in Russian]. *Proceedings of Irkutsk State Technical University*; 2015; 8: 47-51.
-

About the authors

Leonid Vadimovich Arshinskiy (b. 1957) graduated from the Irkutsk State University (Irkutsk-city) in 1979, Dr of Tech. Sc. (2008). He is Head of the Department "Information Systems and Information Security" Irkutsk State Transport University. Corresponding Member of the Russian Academy of Natural Sciences and Russian Engineering Academy. He is author of more than 200 scientific articles and abstracts in the field of aircraft with ground effect wings, image recognizing, plausible inference, aggregate evaluation of systems etc. AuthorID (RCI): 520252; Author ID (Scopus): 57193195356; Researcher ID (WoS): C-3869-2013; ORCID 0000-0001-5135-7921. larsh@mail.ru.

Ermakov Anatoly Anatolyevich (b. 1951). He graduated from the Kiev Higher Military Aviation Engineering School in 1976, Ph.D. (1982). Professor, Department "Information Systems and Information Security", Irkutsk State Transport University. The list of scientific papers includes about 70 works in the field of technical diagnostics, reliability and operation of complex systems, pattern recognition, etc. AuthorID (RCI): 861476; Researcher ID (WoS): AAH-4024-2020; ORCID 0000-0002-9312-9916. anatolij.ermackow2011@yandex.ru.

Nitezuk Marina Sergeevna (b. 1984) graduated from the Irkutsk State Transport University in 2010. Senior lecturer and graduate student of the department "Information Systems and Information Security" of Irkutsk State Transport University. The list of scientific works includes 10 works in the field of information systems and technologies. AuthorID (RCI): 1007958; Author ID (Scopus): 57214999113; Researcher ID (WoS): AAH-4012-2020; ORCID 0000-0003-3420-8312. marino_@mail.ru.

Received February 10, 2020. Revised March 10, 2020. Accepted March 15, 2020.
