

Алгоритм очевидности для первопорядковой логики предикатов с равенством

Вторушин Ю.И.¹

Статья является продолжением статьи [1]. Рассматривается алгоритм верификации формализованных математических доказательств для логики предикатов первого порядка с равенством. Доказываются теоремы о его корректности и полноте.

Ключевые слова: автоматическое доказательство теорем, система автоматизации дедукции, верификация доказательств, язык первого порядка, исчисление предикатов, продукционная система, искусственный интеллект.

В статье [1] был описан алгоритм очевидности для логики предикатов без равенства, который эффективно решает задачу верификации формализованных математических текстов. Данная статья является продолжением статьи [1]. В настоящей работе рассматривается алгоритм очевидности для логики предикатов с равенством. При этом используются принятые в [1, 2, 3] понятия и обозначения.

Кратко напомним обозначения. В работе [1] для $i = 1, 2, 3$ рассмотрены системы натуральной дедукции \mathcal{D}_i и алгоритмы очевидности \mathcal{E}_i , которые формулируются с помощью продукционных систем \mathcal{V}_i . При этом случай $i = 1$ соответствует случаю классической логики, $i = 2$ — интуиционистской, а $i = 3$ — минимальной. В данной статье для каждого $i = 1, 2, 3$ формулируется алгоритм очевидности \mathcal{E}_i и продукционная система \mathcal{V}_i , которые соответственно включают \mathcal{E}_i и \mathcal{V}_i как часть. Все положительные свойства алгоритма \mathcal{E}_i наследуются алгоритмом \mathcal{E}_i .

Отметим, что системы натуральной дедукции \mathcal{D}_i рассмотрены в статье [2]. Исчисление \mathcal{D}_i получается из \mathcal{D}_i в результате добавления аксиомы $t = t$ и правила вывода

$$\frac{t = s; \quad \Phi(t)}{\Phi(t//s)},$$

¹*Вторушин Юрий Игоревич* — ведущий программист компании DSS Lab, e-mail: urchick@mail.ru.

Vtorushin Yuriy Igorevich — leader programmer, DSS Lab.

где через $\Phi(t//s)$ обозначена формула, которая может быть получена из формулы Φ в результате замены некоторых вхождений (быть может, всех) термина t на терм s .

Рассматриваемый алгоритм очевидности \mathfrak{E}_i решает задачу проверки правильности формальных доказательств, записанных в стиле обычных математических статей, и может быть реализован в виде эффективной процедуры. Для его обоснования в этой статье формулируются те же теоремы, что и в статье [1], только сейчас речь идет не о \mathcal{D}_i , \mathcal{V}_i и \mathcal{E}_i , а о \mathfrak{D}_i , \mathfrak{V}_i и \mathfrak{E}_i .

Пусть псевдоформула Φ обладает свойством чистоты переменных [3]. Пусть r_1, r_2, \dots, r_k есть список некоторых попарно непересекающихся вхождений псевдотермов в Φ , которые не содержат связанных переменных. Определим операцию *замены* псевдотермов r_1, \dots, r_k на псевдотерм t . Результат ее будет обозначаться символом

$$\left(\Phi \frac{r_1, r_2, \dots, r_k}{t} \right). \quad (1)$$

Рассмотрим псевдоформулу Ψ такую, что $\Phi = \Psi \begin{pmatrix} x_1 & \dots & x_k \\ r_1 & \dots & r_k \end{pmatrix}$, где каждая свободная предметная переменная x_i входит в Ψ ровно один раз. Тогда (1) есть по определению псевдоформула $\Psi \begin{pmatrix} x_1 & \dots & x_k \\ t & \dots & t \end{pmatrix}$.

Продукционная система \mathfrak{V}_i получается из системы \mathcal{V}_i в результате добавления новых примитивных задач и новых правил декомпозиции. Кроме примитивных задач продукционной системы \mathcal{V}_i система \mathfrak{V}_i включает также примитивные задачи следующего вида:

$$\Gamma \triangleright (t = t).$$

По определению будем считать, что задачи вида $\Gamma \triangleright (t_1 = t_2)$ примитивизируются подстановками θ вида $\mathbf{MGU}(t_1, t_2)$, если псевдотермы t_1 и t_2 унифицируемы.

Кроме семи правил декомпозиции задач системы \mathcal{V}_i продукционная система \mathfrak{V}_i включает следующие два правила декомпозиции.

8. Пусть s_1, \dots, s_n — список всех тех псевдотермов псевдоформулы Ψ , которые унифицируются с псевдотермом s . Пусть $\sigma_i = \mathbf{MGU}(s, s_i)$, вхождения в Ψ псевдотермов s_{i_1}, \dots, s_{i_k} попарно не пересекаются и $\sigma_{i_1 \dots i_k}$ есть комбинация подстановок $\sigma_{i_1}, \dots, \sigma_{i_k}$. Тогда если разрешима задача $\Gamma \triangleright \left(\Psi \frac{s_{i_1}, \dots, s_{i_k}}{t} \right) \sigma_{i_1 \dots i_k}$ и θ — ее допустимая подстановка, то задача $\langle \alpha, t = s \rangle \Gamma \triangleright \Psi$ также считается разрешимой и комбинация подстановок

θ и $\sigma_{i_1 \dots i_k}$ считается ее допустимой подстановкой:

$$\frac{\Gamma \triangleright \left(\Psi \frac{s_{i_1}, s_{i_2}, \dots, s_{i_k}}{t} \right) \sigma_{i_1 i_2 \dots i_k}}{\langle \alpha, t = s \rangle \Gamma \triangleright \Psi}.$$

9. Учитывая, что предикат равенства обладает свойством симметричности, мы получаем еще одно правило:

$$\frac{\Gamma \triangleright \left(\Psi \frac{s_{i_1}, s_{i_2}, \dots, s_{i_k}}{t} \right) \sigma_{i_1 i_2 \dots i_k}}{\langle \alpha, s = t \rangle \Gamma \triangleright \Psi}.$$

Формулировка продукционных систем \mathfrak{V}_i завершена. Каждая продукционная система \mathfrak{V}_i задает алгоритм поиска решения дедуктивной задачи без уточнения стратегии построения дерева поиска типа И/ИЛИ. Зафиксировав конкретную стратегию, мы получим некоторую реализацию так сформулированного алгоритма. Напомним, что стратегия построения дерева поиска заключается в задании способа выбора листа для формирования у него, во-первых, множества всех примитивизирующих этот лист подстановок, во-вторых, всех дочерних связей вершин, соответствующих правилам декомпозиции. Затем каждая примитивизирующая подстановка “поднимается” по соединяющей лист с корнем ветви с помощью операции комбинации [3]. Тем самым в ходе процесса построения дерева поиска у его вершин формируются наборы допустимых подстановок. В качестве символических названий сформулированных правил декомпозиции 8 и 9 будем соответственно использовать обозначения $(LR = \triangleright)$ и $(RL = \triangleright)$.

Теорема. *Алгоритм проверки непосредственных обоснований всегда завершает свою работу.*

Доказательство. Обоснование этого утверждения для \mathfrak{V}_i почти не отличается от доказательства [1] такой же теоремы, сформулированной для \mathcal{V}_i . Пусть n есть общее число вхождений всех подформул в исходную задачу

$$\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi. \quad (2)$$

Чтобы установить применимость алгоритма к любой задаче (2) покажем, что дерево поиска конечно.

Действительно, применение правил декомпозиции $(\triangleright \&)$, $(\triangleright \vee)$, $(\triangleright \exists)$, $(\vee \triangleright)$, $(\& \triangleright)$, $(\supset \triangleright)$, $(\forall \triangleright)$, $(LR = \triangleright)$, $(RL = \triangleright)$ всегда приводит к тому, что значение параметра n у каждой дочерней задачи становится меньше значения этого параметра у родительской задачи. Отсюда следует, что высота дерева поиска задачи (2) не превосходит n . Таким образом, поисковое дерево конечно. \square

Изменив в алгоритме очевидности \mathcal{E}_i обработку непосредственных обоснований в соответствии с продукционной системой \mathfrak{V}_i , мы получаем алгоритм очевидности \mathfrak{E}_i для первопорядковой логики предикатов с равенством. При этом новая процедура \mathfrak{E}_i наследует все положительные свойства процедуры \mathcal{E}_i .

Очевидным следствием теоремы о конечности поисковых деревьев у продукционной системы \mathfrak{V}_i и формулировки алгоритма очевидности \mathfrak{E}_i является следующее утверждение.

Следствие. *Алгоритм очевидности \mathfrak{E}_i всегда завершает свою работу.*

В качестве примера рассмотрим формальное доказательство теоремы Пифагора “если $\vec{a} \cdot \vec{b} = 0$, то $(\vec{a} + \vec{b})^2 = \vec{a}^2 + \vec{b}^2$ ”. Такое доказательство в [2] раздроблено до уровня применений правил вывода системы натуральной дедукции \mathfrak{D}_i . По сравнению с [2] алгоритм очевидности \mathfrak{E}_i позволяет записать это доказательство в стиле обычной математической статьи:

```

environ
  type Real;
  func s(Real, Real) -> Real;
  func o() -> Real;
  reserve x,y for Real;
  1: for x holds s(x,o()) = x;
  2: for x,y holds s(x,y) = s(y,x);

  type Vector;
  func f(Vector, Vector) -> Vector;
  func h(Vector,Vector) -> Real;
  reserve u,v,w for Vector;
  3: for u,v,w holds h(f(u,v),w) = s(h(u,w),h(v,w));
  4: for u,v,w holds h(u,f(v,w)) = s(h(u,v),h(u,w));
  5: for u,v holds h(u,v) = h(v,u);
text

for a,b being Vector st h(a,b) = o()
  holds h(f(a,b), f(a,b)) = s(h(a,a), h(b,b))
proof
  let a be Vector, b be Vector;
  assume 6: h(a,b) = o();
  7: s(o(), h(b,b)) = s(h(b,b), o()) by 2
      . = h(b,b) by 1;
  thus h(f(a,b), f(a,b)) = s(h(a, f(a,b)), h(b, f(a,b))) by 3
      . = s(s(h(a,a), h(a,b)), h(b, f(a,b))) by 4
      . = s(s(h(a,a), o()), h(b, f(a,b))) by 6
      . = s(h(a,a), h(b, f(a,b))) by 1
      . = s(h(a,a), s(h(b,a), h(b,b))) by 4
      . = s(h(a,a), s(o(), h(b,b))) by 6,5
      . = s(h(a,a), h(b,b)) by 7;
end;

```

В этом доказательстве рассмотрим более подробно непосредственное обоснование

$$s(h(a, a), s(h(b, a), h(b, b))) = s(h(a, a), s(o(), h(b, b))) \text{ by 6,5.}$$

Для проверки этого утверждения алгоритм очевидности \mathfrak{E}_i сформирует дерево поиска типа И/ИЛИ, в котором содержится соответствующее допустимой подстановке $\theta = \begin{pmatrix} U & V \\ a & b \end{pmatrix}$ следующее имеющее вид только одной ветви деривационное поддерево:

$$\begin{array}{c} \text{Yes} \\ \uparrow \lambda \\ \triangleright s(h(a, a), s(h(a, b), h(b, b))) = s(h(a, a), s(h(a, b), h(b, b))) \\ \uparrow \nu \\ \langle 6, h(a, b) = o() \rangle \triangleright \left(\begin{array}{l} s(h(a, a), s(h(a, b), h(b, b))) \\ = s(h(a, a), s(o(), h(b, b))) \end{array} \right) \\ \uparrow \mu \\ \left. \begin{array}{l} \langle 5, h(U, V) = h(V, U) \rangle \\ \langle 6, h(a, b) = o() \rangle \end{array} \right\} \triangleright \left(\begin{array}{l} s(h(a, a), s(h(b, a), h(b, b))) \\ = s(h(a, a), s(o(), h(b, b))) \end{array} \right) \\ \uparrow \\ \left. \begin{array}{l} \langle 5, \forall v (h(U, v) = h(v, U)) \rangle \\ \langle 6, h(a, b) = o() \rangle \end{array} \right\} \triangleright \left(\begin{array}{l} s(h(a, a), s(h(b, a), h(b, b))) \\ = s(h(a, a), s(o(), h(b, b))) \end{array} \right) \\ \uparrow \\ \left. \begin{array}{l} \langle 5, \forall u \forall v (h(u, v) = h(v, u)) \rangle \\ \langle 6, h(a, b) = o() \rangle \end{array} \right\} \triangleright \left(\begin{array}{l} s(h(a, a), s(h(b, a), h(b, b))) \\ = s(h(a, a), s(o(), h(b, b))) \end{array} \right) \end{array}$$

Лист этого дерева примитивизируется подстановкой $\lambda = \varepsilon$ и получается из корня дерева поиска в результате следующей цепочки применений правил декомпозиции продукционной системы \mathfrak{A}_i : $(\forall \triangleright)$, $(\forall \triangleright)$, $(LR = \triangleright)$, $(LR = \triangleright)$. Комбинация подстановок λ , $\nu = \varepsilon$ и $\mu = \begin{pmatrix} U & V \\ a & b \end{pmatrix}$ дает допустимую подстановку θ для корня дерева поиска.

Обоснование теоремы о корректности для \mathfrak{A}_i получается из доказательства такой же теоремы в [1], сформулированной для \mathcal{V}_i . В связи с этим имеет смысл рассмотреть только примитивные задачи и правила декомпозиции, которых нет в \mathcal{V}_i .

Теорема о корректности. *Если задача $\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi$ разрешима в \mathfrak{A}_i , то существует цепочка непосредственных обоснований, которая устанавливает выводимость формулы Ψ из формул Φ_1, \dots, Φ_m в \mathfrak{D}_i .*

Доказательство. Пусть θ есть подстановка из набора допустимых подстановок задачи $P = (\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi)$. Мы предполагаем, что

подстановка θ является основной — в противном случае мы можем заменить “нерешенные” метапеременные локальными константами. Пусть Θ является соответствующим этой подстановке θ деривационным деревом. Будем обозначать штрихом следующее соответствие: $\Phi' = \Phi\theta$. Индукцией по высоте дерева Θ покажем, как в рамках \mathfrak{D}_i построить цепочку непосредственных обоснований, которая устанавливает выводимость формулы Ψ' из формул Φ'_1, \dots, Φ'_m в \mathfrak{D}_i . В рамках \mathfrak{D}_i рассмотрим только те варианты порождения дерева Θ , которых нет в [1].

Пусть P примитивизируется подстановкой θ . Тогда $\Psi = (t_1 = t_2)$, $\Psi' = (t'_1 = t'_2)$ и $t'_1 = t'_2$. В этом случае считается, что в \mathfrak{D}_i формула Ψ' является непосредственным следствием формул Φ'_1, \dots, Φ'_m .

Пусть $P = (\langle \alpha_i, t = s \rangle \Pi \triangleright \Psi)$ получается по правилу ($LR = \triangleright$), а дочерняя задача $\Pi \triangleright \Upsilon\sigma_{i_1\dots i_k}$ является корнем поддерева Θ_1 , где $\Upsilon = \left(\Psi \frac{s_{i_1}, \dots, s_{i_k}}{t} \right)$. Заметим, что $\Upsilon' = \Upsilon\theta = (\Upsilon\sigma_{i_1\dots i_k})\theta$, $\Upsilon = \Upsilon(t)$, $\Upsilon' = (\Upsilon(t))' = \Upsilon'(t')$ и Ψ' имеет вид $\Upsilon'(t'/s')$. Пусть по индуктивному предположению деривационному поддереву Θ_1 , соответствующему задаче $\Pi \triangleright \Upsilon\sigma_{i_1\dots i_k}$, сопоставляется цепочка непосредственных обоснований

$$\Delta(\Upsilon' \text{ by } \gamma_1, \dots, \gamma_n).$$

Указанная цепочка устанавливает выводимость Υ' из Φ'_1, \dots, Φ'_m в \mathfrak{D}_i . Тогда дереву Θ мы можем сопоставить цепочку непосредственных обоснований

$$(\alpha_i : t' = s') \Delta(\beta : \Upsilon' \text{ by } \gamma_1, \dots, \gamma_n) (\Upsilon'(t'/s') \text{ by } \alpha_i, \beta).$$

Полученная цепочка устанавливает выводимость Ψ' из Φ'_1, \dots, Φ'_m в \mathfrak{D}_i .

Аналогично рассматривается случай ($RL = \triangleright$). Если P имеет вид $(\langle \alpha_i, s = t \rangle \Pi \triangleright \Psi)$, то для обоснования выводимости Ψ' из Φ'_1, \dots, Φ'_m требуется более длинная цепочка непосредственных умозаключений:

$$\begin{aligned} & (\alpha_i : s' = t') (\gamma : s' = s') \\ & (\delta : t' = s' \text{ by } \alpha_i, \gamma) \Delta(\beta : \Upsilon' \text{ by } \gamma_1, \dots, \gamma_n) (\Upsilon'(t'/s') \text{ by } \delta, \beta). \end{aligned}$$

□

Корректность алгоритма очевидности \mathfrak{E}_i является следствием корректности \mathfrak{B}_i . Это нетрудно обосновать с помощью тех же рассуждений, что и в [1], в связи с чем мы доказательство опускаем.

Следствие. Если алгоритм очевидности \mathfrak{E}_i распознал, что “ Δ есть обоснование формулы Ψ с посылками Φ_1, \dots, Φ_m ”, то существует натуральный вывод Δ' в \mathfrak{D}_i формулы Ψ из посылок Φ_1, \dots, Φ_m .

Полнота \mathfrak{B}_i устанавливается также как и в [1].

Теорема о полноте. Если формула Ψ является непосредственным следствием формул Φ_1, \dots, Φ_m в системе натуральной дедукции \mathfrak{D}_i , то задача $\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi$ разрешима в \mathfrak{V}_i .

Доказательство. Пусть формула Ψ получает обоснование с помощью одного из прямых правил вывода из формул Φ_1, \dots, Φ_m :

$$(\alpha_1 : \Phi_1) \dots (\alpha_m : \Phi_m) \Delta (\Psi \text{ by } \alpha_1, \dots, \alpha_m).$$

Рассмотрим только те варианты непосредственных обоснований в \mathfrak{D}_i формулы Ψ из формул Φ_1, \dots, Φ_m , которые отсутствуют в \mathfrak{D}_i .

Пусть формула Ψ является аксиомой $t = t$. Тогда задача

$$\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi$$

является примитивной. Следовательно, она разрешима в \mathfrak{V}_i .

Пусть формула $\Psi = \Upsilon(t/s)$ получает обоснование с помощью правила ($=v$) из формул $t = s$ и $\Upsilon(t)$:

$$(\alpha_1 : t = s) (\alpha_2 : \Upsilon(t)) \Delta (\Upsilon(t/s) \text{ by } \alpha_1, \alpha_2).$$

Тогда задача $(\langle \alpha_1, t = s \rangle \langle \alpha_2, \Upsilon(t) \rangle \Gamma \triangleright \Upsilon(t/s))$ разрешима в \mathfrak{V}_i . Действительно, в этом случае $\Upsilon(t) = \left(\Upsilon(t/s) \frac{s, s, \dots, s}{t} \right) \varepsilon$ и, таким образом, поисковое дерево содержит соответствующее допустимой подстановке ε следующее деривационное поддерево, которое возникает в результате применения правила декомпозиции ($LR = \triangleright$):

$$\begin{array}{c} \langle \alpha_2, \Upsilon(t) \rangle \Gamma \triangleright \Upsilon(t) \\ \uparrow \\ \langle \alpha_1, t = s \rangle \langle \alpha_2, \Upsilon(t) \rangle \Gamma \triangleright \Upsilon(t/s) \end{array}$$

□

Также как в [1] в результате получаем полноту алгоритма очевидности \mathfrak{E}_i .

Следствие. Если Δ есть формальное доказательство формулы Ψ из посылок Φ_1, \dots, Φ_m в \mathfrak{D}_i , то алгоритм очевидности \mathfrak{E}_i установит, что “ Δ есть обоснование формулы Ψ с посылками Φ_1, \dots, Φ_m ”.

Список литературы

- [1] Вторушин Ю.И., “О верификации формализованных математических доказательств”, *Интеллектуальные системы*, **24**:1 (2020), 7–24.
- [2] Вторушин Ю.И., “О поиске вывода в системе натуральной дедукции логики предикатов”, *Интеллектуальные системы*, **13**:3 (2009), 263–288.
- [3] Охотников О.А., “О поиске натурального классического логического вывода с использованием частичной скульемизации”, *Интеллектуальные системы*, **23**:4 (2019), 39–90.

References

- [1] Vtorushin Yu.I., “About verification of formalized mathematical proofs (in Russian)”, *Intelligent systems*, **24**:1 (2020), 7–24.
- [2] Vtorushin Yu.I., “O poiske vyvoda v sisteme natural’noj dedukcii logiki predikatov (in Russian)”, *Intelligent systems*, **13**:3 (2009), 263–288.
- [3] Okhotnikov O.A., “About proof-search in classical natural deduction calculus using partial skolemization (in Russian)”, *Intelligent systems*, **23**:4 (2019), 39–90.

Evidence algorithm for first-order logic with equality Vtorushin Yu.I.

The article is a continuation of the article [1]. An algorithm for verifying formalized mathematical proofs for first-order predicate logic with equality is considered. Theorems about its correctness and completeness are proved.

Keywords: automated theorem proving, system for automated deduction, first order language, predicate calculus, production system, artificial intelligence.