ЭФФЕКТИВНЫЙ АЛГОРИТМ ОПРЕДЕЛЕНИЯ ИСТИННОСТИ УТВЕРЖДЕНИЙ О ДЕЙСТВИТЕЛЬНЫХ ЧИСЛАХ В СИГНАТУРЕ ОТНОШЕНИЙ ПОРЯДКА

Коварцев А.Н.

Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет) (СГАУ)

Аннотация

В статье предлагается новый эффективный алгоритм определения истинности утверждений о действительных числах в сигнатуре отношений порядка. В отличие от известного алгоритма А. Тарского, предложенный алгоритм сводит переборную задачу проверки истинности любого утверждения о вещественных числах к оптимизационной задаче. В новой версии алгоритма могут использоваться не только алгебраические, но и трансцендентные функции.

Ключевые слова: разрешимость предиката, теорема Тарского, замкнутая формула, доказательство истинности, глобальная оптимизация, сложность алгоритмов.

Введение

Многие практические приложения теоретической информатики могут быть сформулированы в виде некоторого утверждения $P(x_1, ..., x_n)$, которое, в зависимости от значений набора переменных, выдаёт результативный ответ (**И** или **Л**). Естественно, что на содержательном уровне нас интересует не сама постановка задачи как таковая, а существование алгоритма, устанавливающего истинность предиката P, т.е. проблема *разрешимости* предиката [2].

Имеется множество примеров разрешимости формальных систем [2]. В частности, алгоритмически *разрешима* проблема проверки истинности любого утверждения про конечное число вещественных чисел с отношениями равенства и порядка [3]. Как известно, первый такой алгоритм предложил А. Тарский в 1948 году. Однако алгоритм Тарского оказался настолько неэффективным, что практически воспользоваться им невозможно.

В данной работе предлагается новый, более эффективный алгоритм проверки истинности утверждений, заданных на множестве действительных чисел в сигнатуре отношений порядка, сводящийся к решению оптимизационной задачи.

Следует отметить, что под *эффективностью* здесь понимается не достижение полиномиальной сложности алгоритма, а возможность его практического использования для решения широкого круга прикладных задач. Естественно, что сложность такого алгоритма должна быть меньше экспоненциальной.

В качестве практически значимого примера использования предложенного алгоритма можно привести широкий круг задач, связанных с верификацией вычислительных программ, т.е. программ, реализующих вычислительные методы.

1. Постановка задачи

Рассмотрим утверждения исчисления предикатов первого порядка с сигнатурой, включающей отношения порядка. Более точно рассмотрим сигнатуру, содержащую предикаты $=, \le, <, >, \ge, \ne$, арифметические операции и аналитические функции над действительными числами. Предикаты и операции понимаются естественным образом. В этом случае произвольный предикат можно представить в виде:

$$P_\theta(X) \equiv f(X)\ \theta\ 0, \qquad (1)$$

где θ – любое из перечисленных выше отношений порядка, $f(X)$ – аналитическая функция (алгебраическая или трансцендентная или их комбинация), $X = (x_1, ..., x_n)'$ – переменные предиката.

Поставим задачу разработки эффективного алгоритма определения истинности замкнутой формулы, т.е. формулы, не имеющей свободных параметров. Например, аксиомы порядка:

$$\forall x\ \forall y\ (((x \le y) \wedge (y \le x)) \to (x = y)).$$

2. «Решающая» функция

Каждому предикату (1) можно поставить в соответствие множество вида:

$$M_\theta = \{X \mid f(X)\ \theta\ 0\}.$$

В качестве *базисных отношений* порядка выберем отношения $\le$ и $<$. Очевидно, что через базисные отношения можно выразить все остальные отношения порядка:

$$P_> = \overline{P}_\le,$$
$$P_\ge = \overline{P}_<,$$
$$P_= = \overline{P}_< \wedge P_\le,$$
$$P_\ne = \overline{P}_\le \vee P_<.$$

Введём функцию $\phi(X) = -f(X)$ и через неё определим дополнение базисных множеств соответствующих базисных отношений:

$$\overline{M}_\le = \{X \mid \phi(X) < 0\} \text{ и } \overline{M}_< = \{X \mid \phi(X) \le 0\}.$$

Между логическими операциями над предикатами и теоретико-множественными операциями над множествами существует очевидная связь:

$$\overline{P(X)} \Leftrightarrow \overline{\{X \mid f(X) \le 0\}} = \{X \mid \phi(X) < 0\}$$
$$\overline{\{X \mid f(X) < 0\}} = \{X \mid \phi(X) \le 0\},$$

или

$$P(X) \wedge Q(X) \Leftrightarrow \{X \mid f_P(X) \le 0\} \bigcap \{X \mid f_Q(X) \le 0\},$$
$$P(X) \vee Q(X) \Leftrightarrow \{X \mid f_P(X) \le 0\} \bigcup \{X \mid f_Q(X) \le 0\}.$$

Для описания логических операций над предикатами введём «решающую» функцию $F(X)$ [1]. Для этого первоначально определим «индикаторные» функции следующим образом:

1) для отношения $\leq$

$$\beta_{\leq}(X)=\begin{cases}1 & \text{если } f(X)>0,\\ 0 & \text{если } f(X)\leq 0;\end{cases} \qquad (2)$$

2) для отношения $<$

$$\beta_{<}(X)=\begin{cases}1 & \text{если } f(X)\geq 0,\\ 0 & \text{если } f(X)<0.\end{cases} \qquad (3)$$

Тогда для логической операции конъюнкции «решающую» функцию можно определить следующим образом:

$$F_{\wedge}(X)=(\beta_P f_P+\beta_Q f_Q)/(\beta_P+\beta_Q+\overline{\beta}_P\overline{\beta}_Q)+ \\ +\overline{\beta}_P\overline{\beta}_Q(f_P+f_Q)/2. \qquad (4)$$

В формуле (7) подразумевается, что индикаторные функции (5)–(6) и функции f_p и f_Q зависят от X.

«Решающую» функцию для операции дизъюнкции определим так:

$$F_{\vee}(X)=(\overline{\beta}_P f_P+\overline{\beta}_Q f_Q)/(\overline{\beta}_P+\overline{\beta}_Q+\beta_P\beta_Q)+ \\ +\beta_P\beta_Q(f_P+f_Q)/2. \qquad (5)$$

«Решающая» функция атомарных формул, использующих базисные отношения ($P_{\leq}(X)\equiv f(X)\leq 0$ или $P_{<}(X)\equiv f(X)<0$), определяется простым выражением:

$$F(X)=f(X).$$

Тогда решающую функцию для операции отрицания можно определить следующим образом: $\overline{P}_{\leq}=P_{>}\phi(X)<0$, откуда $F(X)=-f(X)$. Аналогично для отношения строгого порядка $<$:

$\overline{P}_{<}=P_{\geq}\equiv\phi(X)\leq 0$, и в результате $F(X)=-f(X)$.

Далее нам потребуется понятие интерпретации формулы. Определим это понятие так, как это сделано в работе [2].

Оценкой π назовём отображение, которое ставит в соответствие каждой индивидуальной переменной некоторый элемент множества, являющийся носителем интерпретации. Этот элемент будем называть значением переменной при данной оценке и обозначать $[x](\pi)$. Значение атомарной формулы $P(x_1,\ldots,x_n)$ определим как $[P]([x_1](\pi),\ldots,[x_n](\pi))$. Для логических операций справедливо:

$$[\overline{P}](\pi)=\overline{[P]}(\pi),$$
$$[P\wedge Q](\pi)=[P](\pi)\wedge[Q](\pi),$$
$$[P\vee Q](\pi)=[P](\pi)\vee[Q](\pi),$$

где P и Q – формулы.

Очевидно, что атомарные формулы $P_{\leq}(X)$ (или $P_{<}(X)$) истинны для тех оценок интерпретации переменных, для которых «решающая» функция $F(X)\leq 0$ (или $<$) 0.

Теорема 1. Формула $P\wedge Q$, связывающая предикаты, использующие базисные отношения, истинна тогда и только тогда и для тех оценок интерпретации переменных формулы, для которых «решающая» функция (4) меньше или равна нулю.

Доказательство. Несложно показать, что для логической операции конъюнкции, если хотя бы один из предикатов использует отношение $\leq$, то $P\wedge Q\Leftrightarrow\{X|F(X)\leq 0\}$. В противном случае $P\wedge Q\Leftrightarrow\{X|F(X)<0\}$. Пусть $P\Leftrightarrow\{X|f_P(X)\leq 0\}$, а $Q\Leftrightarrow\{X|f_Q(X)<0\}$.

«Решающая» функция (4) по построению предполагает четыре варианта интерпретации переменных:

1) $[X](\pi_1)$: $P(X)=1$, $Q(X)=0$. Из чего следует, что $\beta_P=0$, $\beta_Q=1$, но тогда $F_{\wedge}(X)=f_Q$. Из чего следует, что для оценки π_1 $F_{\wedge}(X)>0$.
2) $[X](\pi_2)$: $P(X)=0$, $Q(X)=1$. Для этой оценки $\beta_P=1$, $\beta_Q=0$ и $F_{\wedge}(X)=f_P$. Из чего следует, что для оценки π_2 $F_{\wedge}(X)>0$.
3) $[X](\pi_3)$: $P(X)=0$, $Q(X)=0$. Для этой оценки $\beta_P=1$, $\beta_Q=1$ и $F_{\wedge}(X)=(f_P+f_Q)/2$, но поскольку $f_p>0$ и $f_q\geq 0$, то для оценки π_3 справедливо $F_{\wedge}(X)>0$.
4) И наконец, существует оценка $[X](\pi_4)$: $P(X)=1$, $Q(X)=1$, для которой $\beta_P=0$, $\beta_Q=0$. В этом случае $F_{\wedge}(X)=(f_P+f_Q)/2$, но поскольку $f_p\leq 0$ и $f_q<0$, то для оценки π_4 $F_{\wedge}(X)\leq 0$.

Но тогда справедливо $P\wedge Q\Leftrightarrow\{X|F(X)\leq 0\}$.

Верно и обратное. Пусть для оценки $[X](\pi_4)$ выполняется условие $F_{\wedge}(X)\leq 0$. Учитывая структуру формулы (4), такая ситуация возможна, если у «решающей» функции используется второе слагаемое, т.е. $F_{\wedge}(X)=\overline{\beta}_P\overline{\beta}_Q(f_P+f_Q)/2$, но это возможно, когда $\beta_P=0$, $\beta_Q=0$, откуда следует, что $P(X)=1$, $Q(X)=1$. Аналогично доказываются и остальные случаи оценок интерпретаций переменных.

Для формул, использующих логическую связку «ИЛИ», можно сформулировать следующую теорему. Приведем её без доказательства.

Теорема 2. Формула $P\vee Q$, связывающая предикаты, использующие базисные отношения, истинна тогда и только тогда и для тех оценок интерпретации переменных формулы, для которых «решающая» функция (5) меньше или равна нулю.

Замечание. Для решающей функции $F_{\vee}(X)$ следует использовать отношение порядка $<$, если хотя бы один из предикатов содержит это отношение.

Остальные логические связки можно выразить через логические операции «И», «ИЛИ», «НЕ». Таким образом, для произвольной формулы рекурсивно можно поставить в соответствие «решающую» $F_{\phi}(X)$ функцию, а следовательно, задачу проверки условия:

$$\phi\equiv(F_{\phi}(X)\leq(\text{или }<)\ 0).$$

3. Алгоритм проверки истинности утверждений

Зададимся вопросом, что же нам даёт «решающая» функция? Применительно к задаче проверки истинности формул для случая, когда носителем интерпретации выступает конечное множество, алгоритм достаточно прост. Можно построить таблицу истинности формулы (на конечном множестве оценок) и этим закрыть проблему. В случае, когда носителем выступают действительные числа, задача значительно осложняется, поскольку число оценок становится бесконечным и область переменных формулы «раскрашивается» в чёрно-белые тона (**И** или **Л**). Кроме того, зная значение интерпретации формулы для какой-либо оценки (например, **Л**), мы не можем знать, в каком направлении изменения переменных следует искать истинную оценку интерпретации переменных.

«Решающая» функция в пространстве переменных предиката устанавливает порядок (рис. 1), в связи с чем появляется возможность организации поиска таких сочетаний переменных $F(X)$, для которых она принимает отрицательные значения. Для этого достаточно поставить задачу поиска глобального минимума «решающей» функции:

$$X^* = \arg\min_X F(X).$$

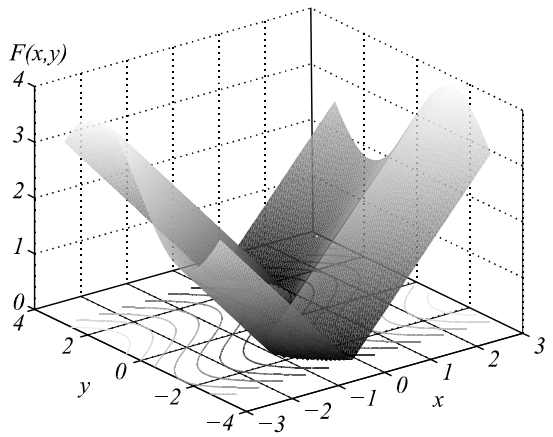


Рис. 1. Общий вид «решающей» функции

Для формулы ϕ, имеющей одно свободное вхождение переменной *x*, проверка её истинности эквивалентна оценке истинности формулы $\exists x F_\phi(x)$, которая, в свою очередь, однозначно связана с проверкой условия

$$F_\phi(\arg\min_x F_\phi(x)) < 0, \qquad (6)$$

т.е. решением задачи глобальной оптимизации (6) для соответствующей «решающей» функции.

Обобщая полученные результаты, несложно построить алгоритм проверки истинности утверждений о действительных числах в сигнатуре отношений порядка.

1. Предварительно исходную замкнутую формулу необходимо привести к эквивалентной ей предварённой (пренексной) нормальной форме [2]:

$$\phi \equiv \mathrm{K}_1 x_1 \ldots \mathrm{K}_n x_n \; M(x_1, \ldots, x_n), \qquad (7)$$

где $\mathrm{K}_i \in \{\forall, \exists\}$. Этого несложно добиться за счёт продвижения знака отрицания до атома, переименования переменных и вынесения кванторов в левую часть формулы [2]. При этом мы предполагаем, что в бескванторной части формулы используются только логические связки «И», «ИЛИ», «НЕ».

2. Для бескванторной части формулы построим «решающую» функцию $F_M(X)$.

3. Последовательно, шаг за шагом будем избавляться от связанных переменных в формуле (7), организовав рекурсивный процесс решения оптимизационных задач:

$$\begin{cases} \phi_n = \mathrm{K}_n x_n \; M(x_1, \ldots, x_n), \\ \phi_{n-1} = \mathrm{K}_{n-1} x_{n-1} \; \phi_n(x_1, \ldots, x_{n-1}), \\ \\ \phi = \mathrm{K}_1 x_1 \; \phi_2(x_1). \end{cases} \qquad (8)$$

Для каждой из задач (8) $\phi_n = \mathrm{K}_i x_i \phi_{i+1}(x_1, \ldots, x_i)$ строится «решающая» функция $\mathrm{F}_{\phi_i}(x_1, \ldots, x_i)$ и ищется её глобальный минимум. Причём если $\mathrm{K}_i = \exists$, то $F_{\phi_i}(x_1, \ldots, x_{i-1}) = \min_{x_i} \phi_{i+1}(x_1, \ldots, x_i)$. В случае, если $\mathrm{K}_i = \forall$, то предварительно производится эквивалентное преобразование $\forall x \, \phi = \overline{\exists x \, \bar{\phi}}$ и задача сводится к предыдущему случаю.

4. Примеры

Алгоритм Тарского [3] позволяет установить истинность или ложность утверждений в элементарной теории действительных чисел с операциями сложения и умножения, т.е. для предикатов, использующих алгебраические функции. Рассмотрим, как работает наш алгоритм для предикатов, содержащих тригонометрические функции.

Пример 1. Проверка формулы $\forall x \exists y (\sin x = y)$.

Прежде всего, преобразуем бескванторную часть формулы к нормализованному виду

$$M(x.y) = (\sin x = y) \equiv (\sin x - y = 0) \equiv$$
$$\equiv \overline{(\sin x - y < 0)} \wedge (\sin x - y \le 0) \equiv$$
$$\equiv (y - \sin x \le 0) \wedge (\sin x - y \le 0).$$

Для (7) построим «решающую» функцию, положив $\theta = y - \sin x$ и $f = \sin x - y$, тогда в соответствии с (4) имеем

$$F_M(x, y) = (\beta_\theta \theta + \beta_f f) / (\beta_\theta + \beta_f + \bar{\beta}_\theta \bar{\beta}_f) + \\ + \bar{\beta}_\theta \bar{\beta}_f (\theta + f) / 2. \qquad (9)$$

На первом шаге свернём переменную *y*:

$$\phi_1(x) = \exists y \, (F_M(x, y) \le 0) \equiv (\min_y F_M(x, y) \le 0).$$

Следующим шагом свернём переменную *x*:

$$\phi = \forall x \, (\min_y F_M(x, y) \le 0) \equiv \overline{\exists x \, \overline{(\min_y F_M(x, y) \le 0)}} \equiv$$
$$\equiv \overline{\exists x \, (-\min F_M(x, y) < 0)} \equiv \overline{\min_x (\max_y F_M(x, y)) < 0} \equiv$$
$$\equiv \min_x (\max_y F_M(x, y)) \ge 0.$$

В результате задача проверки истинности замкнутой формулы свелась к оптимизационной задаче и проверке условия

$$(\min_x \max_y F_M(x, y) \geq 0).$$

Вычислительные эксперименты показали, что при изменении переменной $x \in [-3; 3]$ функция $F_1(x) = \max_y F_M(x, y) \equiv 0$, следовательно, исходная формула верна, по крайней мере, в рассматриваемом диапазоне изменения переменной x.

Пример 2. Проверка невыполнимой формулы $\forall y \exists x (\sin x = y)$.

Для данной формулы мы имеем аналогичную предыдущему случаю «решающую» функцию (9).

На первом шаге свернём переменную x:

$$\phi_1(y) = \exists x \, (F_M(x, y) \leq 0) \equiv (\min_x F_M(x, y) \leq 0).$$

Следующим шагом свернём переменную y:

$$\phi = \forall y \, (\min_x F_M(x, y) \leq 0) \equiv \overline{\exists y \, (\overline{\min_x F_M(x, y) \leq 0})} \equiv$$

$$\equiv \overline{\exists y \, (-\min_x F_M(x, y) < 0)} \equiv \overline{\min_y (\max_x F_M(x, y)) < 0)} \equiv$$

$$\equiv \min_y (\max_x F_M(x, y)) \geq 0.$$

Для этого примера функция $F_1(y) = \max_x F_M(x, y)$ показана на рис. 2.

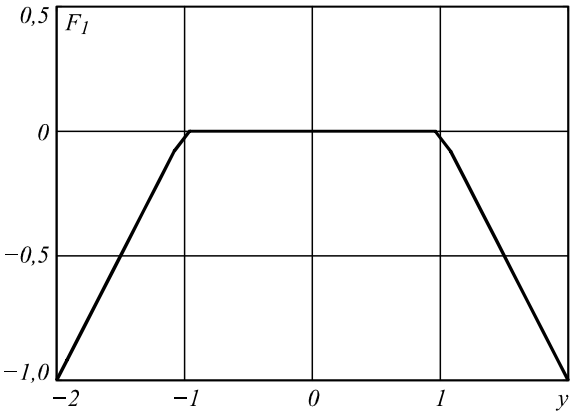


Рис. 2. График «решающей» функции $F_1(y)$

Из рисунка видно, что $\min_y F_1(y)$, по крайней мере, меньше –0,5, следовательно, исходная формула невыполнима.

Заключение

Кратко обсудим полученные результаты.

1. Предложенный выше алгоритм, основанный на использовании «решающей» функции, позволил свести фактически переборную задачу проверки истинности любого утверждения о вещественных числах к *оптимизационной задаче*. Причём в новой версии алгоритма могут использоваться не только алгебраические, но и *трансцендентные* функции.

2. Универсальный алгоритм Тарского оказался чрезвычайно неэффективным. Было доказано [3], что алгоритм Тарского будет работать дважды экспоненциальное время на некоторых «плохих» формулах в зависимости от их длины. Предлагаемый алгоритм, сводящийся к решению задачи глобальной оптимизации, в общем случае также имеет экспоненциальный характер роста его сложности [4], но уже в зависимости от числа оптимизируемых переменных. Размер формулы в этом случае не играет роли.

Кроме того, с точки зрения нахождения приближенного решения оптимизационные задачи имеют более стройные схемы их классификации, чем обычная теория *NP*-полноты. Дело в том, что практическая сложность реализации алгоритмов глобальной оптимизации во многом определяется свойствами оптимизируемой функции.

Например, для класса унимодальных гладких функций алгоритмы оптимизации сходятся к решению за полиномиальное время. В приведённом примере, использующем тригонометрическую функцию, как видно из рис. 1, «решающая» функция имеет множество равнозначных глобальных минимумов, расположенных на дне извилистого оврага. Очевидно, что нахождение любого из них не займёт много времени.

Более того, оптимизационные алгоритмы существенно снижают свою трудоёмкость (вплоть до полиномиального случая) при отказе от поиска точного решения. Известны многочисленные примеры приближённых алгоритмов оптимизации [6], решающих задачу оптимизации за полиномиальное время, к ним же относятся и рандомизированные алгоритмы. В нашем случае в задачах опровержения формулы (пример 2) достаточно найти любое, пусть даже приближённое решение меньше нуля.

3. Предложенный алгоритм имеет вариант практического использования, например, при проверке корректности графа управления сложного программного обеспечения (ПО), в той части, которая связана с проверкой логических выражений, управляющих вычислительным процессом. Такая задача актуальна в процессе тестирования и отладки ПО управления космическими объектами [5].

Благодарности

Работа выполнена при государственной поддержке Министерства образования и науки РФ в рамках реализации мероприятия Программы повышения конкурентоспособности СГАУ среди ведущих мировых научно-образовательных центров на 2013–2020 годы.

Литература

1. **Коварцев, А.Н.** Автоматизация разработки и тестирования программных средств / А.Н. Коварцев. – Самара: Самарский государственный аэрокосмический университет, 1999. – 150 с.
2. **Верещагин, Н.К.** Языки и исчисления / Н.К. Верещагин, А. Шень. – М.: МЦНМО, 2012. – 240 с.
3. **Матиясевич, Ю.В.** Алгоритм Тарского / Ю.В. Матиясевич // Компьютерные инструменты в образовании. – 2008. – № 6. – С. 4-14.

4. **Коварцев, А.Н.** К вопросу об эффективности параллельных алгоритмов глобальной оптимизации функций многих переменных / А.Н. Коварцев, Д.А. Попова-Коварцева // Компьютерная оптика. – 2011. – Т. 35, № 2. – С. 256-261.
5. **Мостовой, Я.А**. Имитационная математическая модель внешней среды в жизненном цикле бортового программного обеспечения управления космической платформой / Я.А. Мостовой // Компьютерная оптика. – 2012. – Т. 36, № 3. – С. 412-418.
6. **Vazirani, Vijay V**. Approximation Algorithms // Vijay V. Vazirani. – Berlin: Springer, 2003. – 375 p.

References

1. **Kovartsev, A.N**. Automating the development and testing of software / A.N. Kovartsev. – Samara: "SSAU" Publisher, 1999. – 150 p. – (In Russian).
2. **Vereshchagin, N.K.** Languages and calculus / N.K. Vereshchagin, A. Shen. – Moscow: "MCCME_PUBL", – 2012. – 240 p. – (In Russian).
3. **Matiyasevich,** Y. Tarski algorithm / Y. Matiyasevich // Computer Tools in Education. – 2008. – Vol. 6. – P. 4-14. – (In Russian).
4. **Kovartsev, A.N** On efficiently of parallel algorithms for global optimization of functions of several variables / A.N. Kovartsev, D.A. Popova-Kovartseva // Computer Optics. – 2011. – Vol. 35(2). – P. 256-261. – (In Russian).
5. **Mostovoi**, **J.A**. Simulation mathematical model of the external ambience in life cycle of on-board software of management cosmic platform / J.A. Mostovoi // Computer Optics. – 2012. – V. 36(3). – P. 412-418.
6. **Vazirani, Vijay V**. Approximation Algorithms // Vijay V. Vazirani. – Berlin: Springer, 2003. – 375 p.

AN EFFICIENT ALGORITHM FOR TESTING THE TRUTH OF ASSERTIONS FOR REAL NUMBERS EXPRESSED IN RELATIONAL SIGNATURES

A.N. Kovartsev
Samara State Aerospace University

Abstract

In this paper a new efficient algorithm is proposed for testing the truth of assertions for real numbers expressed in relational signatures. In contrast to the well known Tarski algorithm, the proposed algorithm reduces the search problem of checking the truth of any assertion for real numbers to the optimization problem. The new version of the algorithm can be used for algebraic and transcendental functions.

Key words: predicate derivation, Tarski theorem, closed formula, the proof of the truth, global optimization, algorithm complexity.

Сведения об авторе



Коварцев Александр Николаевич, 1952 года рождения. В 1975 году окончил Куйбышевский авиационный институт (ныне – Самарский государственный аэрокосмический университет имени академика С.П. Королёва) по специальности «Прикладная математика». Доктор технических наук (2000 год), профессор, работает заведующим кафедрой программных систем СГАУ. Директор школы информатики СГАУ. А.Н. Коварцев – специалист в области автоматизации проектирования, разработки и тестирования сложных программных средств; надёжности программного обеспечения; моделирования параллельных вычислений и глобальной оптимизации. Имеет более 90 научных публикаций.

Alexandr Kovartsev (b. 1952) graduated with honours (1975) from S. P. Korolyov Kuibyshev Aviation Institute (presently, S. P. Korolyov Samara State Aerospace University (SSAU)), majoring in Applied Mathematics. He received his Doctor of Engineering (2000) degrees from Samara State Aerospace University. Professor, chair of SSAU's sub-department software systems SSAU. Director of the School of Computer SSAU. His current research interests include computer-aided design, development and testing of complex software; simulation of parallel computing and global optimization. He has more than 90 scientific publications.

Поступила в редакцию 13 мая 2014 г.