МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ярославский государственный университет им. П.Г.Демидова»

Кафедра компьютерной безопасности и математических методов
обработки информации

Курсовая работа

Алгоритм Тарского. Описание и реализация

Научный руководитель
профессор, д-р ф.-м.н.
_____ В.Г. Дурнев
«___» _____ 2020 г.

Студент группы КБ-41СО
_____ Р. А. Гибадулин
«___» _____ 2020 г.

Ярославль, 2020 г.

Оглавление

Введение

Введение

1 Алгоритм Тарского

Прежде всего определим область математики, истинность утверждений которой должен проверять алгоритм. Затем формально опишем язык, на котором записываются эти утверждения. И, наконец, опишем алгоритм, который по формуле описанного языка строит эквивалентную бескванторную формулу.

1.1 Элементарная алгебра

Под элементарной алгеброй понимается та часть общей теории действительных чисел, в которой используются переменные, представляющие собой действительные числа, и константы для всех рациональных чисел, определены арифметические операции, такие как «сложение» и «умножение», и отношения сравнения действительных чисел – «меньше», «больше» и «равно». То есть рассматриваются системы алгебраических уравнений и неравенств.

Заметим, что используя декартову систему координат, некоторые задачи геометрии можно сформулировать как задачи элементарной алгебры. Например, теорема о пересечении высот треугольника, которая утверждает, что три высоты невырожденного треугольника пересекаются в одной точке, равносильна утверждению: для любых трех точек $A(x_1, y_1)$, $B(x_2, y_2)$ и $C(x_3, y_3)$, не лежащих на одной прямой, существует точка $D(x_4, y_4)$ такая, что $\overrightarrow{AD} \perp \overrightarrow{BC}$, $\overrightarrow{BD} \perp \overrightarrow{AC}$ и $\overrightarrow{CD} \perp \overrightarrow{AB}$. Иначе говоря, если $\overrightarrow{AB} \wedge \overrightarrow{AC} \neq 0$, то система

$$\begin{cases} \left(\overrightarrow{AD}, \overrightarrow{BC}\right) = 0 \\ \left(\overrightarrow{BD}, \overrightarrow{AC}\right) = 0 \\ \left(\overrightarrow{CD}, \overrightarrow{AB}\right) = 0 \end{cases}$$

имеет решение относительно переменных x_4, y_4, где $* \wedge *$ – псевдоскалярное произведение векторов, $(*, *)$ – скалярное произведение векторов.

1.2 Язык элементарной алгебры

Язык элементарной алгебры – это язык логики первого порядка с сигнатурой

$$\tau = \langle\, \mathbb{Q},\, F,\, P,\, \theta,\, \phi \,\rangle,$$

где $\mathbb{Q}$ – множество рациональных чисел, которое является множеством индивидных констант, $F = \{+, \cdot\}$ – множество функциональных символов, $P = \{<, >, =\}$ – множество предикатных символов, $\theta : F \to \mathbb{N}$ такое, что $\theta(+) = 2$ и $\theta(\cdot) = 2$, и $\phi : P \to \mathbb{N}$ такое, что $\phi(<) = 2$, $\phi(>) = 2$ и $\phi(=) = 2$. Из определения отображений θ и ϕ следует, что все $f \in F$ являются двухместными функциональными символами, а все $p \in P$ являются двухместными предикатными символами. Основное множество интерпретации языка L_τ совпадает с множеством действительных чисел $\mathbb{R}$, отображение множества индивидных констант в основное множество определяется естественным образом, так как $\mathbb{Q} \subset \mathbb{R}$, функциональные символы $+$ и $\cdot$ отображаются в сложение и умножение в поле $\mathbb{R}$ соответственно, и предикатные символы $<$, $>$ и $=$ отображаются естественным образом в операции сравнения в $\mathbb{R}$.

Замечание. *Множество констант ограничено рациональными числами лишь потому, что компьютер может быстро работать с ними без потери точности, что нельзя сказать про действительные числа.*

Например, теорема о пересечении высот на языке элементарной алгебры записывается так:

$$(\forall x_1)(\forall y_1)(\forall x_2)(\forall y_2)(\forall x_3)(\forall y_3)$$
$$((\neg((x_2 - x_1)\cdot(y_3 - y_1) - (y_2 - y_1)\cdot(x_3 - x_1) = 0)) \rightarrow$$
$$(\exists x_4)(\exists y_4)(((x_4 - x_3)\cdot(x_2 - x_1) + (y_4 - y_3)\cdot(y_2 - y_1) = 0)\&$$
$$((x_4 - x_2)\cdot(x_1 - x_3) + (y_4 - y_2)\cdot(y_1 - y_3) = 0)\&$$
$$((x_4 - x_1)\cdot(x_3 - x_2) + (y_4 - y_1)\cdot(y_3 - y_2) = 0))).$$

Замечание. *Нет необходимости формальнo вводить такие операции как вычитание, деление и возведение в степень по следующим соображениям:*

$$a - b = a + (-1)\cdot b; \quad \frac{a}{b} > 0 \Leftrightarrow (a > 0\,\&\,b > 0) \vee (a < 0\,\&\,b < 0); \quad x^2 = x \cdot x.$$

1.3 Элиминация кванторов

Определение 1.1. ***Элиминация кванторов*** *– это процесс, порождающий по заданной логической формуле, другую, эквивалентную ей бескванторную формулу, то есть свободную от вхождений кванторов.*

Пусть алгоритм A такой, что $A((Qx)\mathcal{A}) = \mathcal{B}$, где $\mathcal{A}$ и $\mathcal{B}$ – бескванторные формулы языка элементарной алгебры, и формулы $(Qx)\mathcal{A}$ и $\mathcal{B}$ эквивалентны, а Q – квантор. Тогда верно следующее утверждение:

Утверждение 1.1. *Если алгоритм A существует, то существует алгоритм B такой, что для любой формулы $\mathcal{A}$ языка элементарной алгебры $B(\mathcal{A})$ – бескванторная формула, эквивалентная $\mathcal{A}$.*

Доказательство. Определим алгоритм B следующим образом:

- Если $\mathcal{A}$ – бескванторная формула, то $B(\mathcal{A}) = \mathcal{A}$;
- Если $\mathcal{A} = (Qx)\mathcal{B}$, то $B(\mathcal{A}) = A((Qx)B(\mathcal{B}))$. Формула $B(\mathcal{B})$ – бескванторная по построению B, следовательно запись $A((Qx)B(\mathcal{B}))$ корректна, при этом $B(\mathcal{B})$ эквивалентна $\mathcal{B}$, следовательно, $(Qx)\mathcal{B}$ эквивалентна $(Qx)B(\mathcal{B})$, а значит $\mathcal{A}$ эквивалентна $B(\mathcal{A})$. Также заметим, что длина формулы $\mathcal{B}$ строго меньше длины формулы $\mathcal{A}$;
- Если $\mathcal{A}$ не удовлетворяет предыдущим условиям, то
  - либо $\mathcal{A} = \neg\mathcal{B}$, тогда $B(\mathcal{A}) = \neg B(\mathcal{B})$,
  - либо $\mathcal{A} = \mathcal{B} * \mathcal{C}$, тогда $B(\mathcal{A}) = B(\mathcal{B}) * B(\mathcal{C})$, где $* \in \{\vee, \&, \rightarrow\}$.

  При этом длины формул $\mathcal{B}$ и $\mathcal{C}$ меньше длины формулы $\mathcal{A}$.

Алгоритм B определен рекурсивно, при этом на каждом этапе на вход B подаётся формула меньшей длины, следовательно, алгоритм B является конечным, и на каждом шаге выход алгоритма – бескванторная эквивалентная формула. ◀

Замечание. *Данное утверждение верно и для других языков логики предикатов.*

Таким образом, для элиминации кванторов произвольной формулы достаточно построить алгоритм A и применить описанный в утверждении 1.1 алгоритм B.

1.4 Алгоритм Тарского

Термы в языке элементарной алгебры – это многочлены с рациональными коэффициентами от действительных переменных Тогда очевидно, что выражения

$$f < g, \quad f = g, \quad f > g$$

равносильны выражениям

$$f - g < 0, \quad f - g = 0, \quad f - g > 0$$

соответственно, где f и g – термы. Поэтому, не нарушая общности рассуждений, можно считать, что все атомарные формулы имеют вид:

$$f < 0, \quad f = 0, \quad f > 0.$$

Таким образом нас будет интересовать только знак многочлена.

Замечание. *На данном этапе можно считать, что все рассматриваемые многочлены ненулевые, так как знак нулевого многочлена в любой точке определяется тривиальным образом.*

Рассмотрим формулу $\mathcal{A} = (Qx)(f(x)\,\rho\,0)$, где Q – квантор, $f(x)$ – многочлен от одной переменной, ρ – предикат. Известно, что многочлены от одной переменной сохраняют свой знак, то есть значение предиката неравенства или равенства с нулем постоянно, на интервалах между корнями. Таким образом, чтобы знать знак значения многочлена в любой точке, достаточно знать значения многочлена лишь в **конечном** наборе точек – во всех корнях, в каких-то точках между любой парой соседних корней, а также в точках, одна из которых заведомо правее, а другая – левее всех корней. Тогда формуле $\mathcal{A}$ эквивалентна следующая бесквантворная формула:

$$\mathcal{B} = \begin{cases} \bigvee\limits_{x_0 \in X} (f(x_0)\,\rho\,0), & \text{если } Q = \exists \\ \bigwedge\limits_{x_0 \in X} (f(x_0)\,\rho\,0), & \text{если } Q = \forall \end{cases}$$

где X – конечное множество этих точек.

Рассмотрим формулу $\mathcal{A} = (Qx)(\Phi(x))$, где $\Phi(x)$ – бесквантворная формула, которая может содержать вхождения лишь переменной x. Аналогично предыдущему случаю, пусть множество X состоит из корней многочленов, входящих в формулу $\Phi(x)$, и точек, выбранных между парами соседних корней, а также пусть в это множество входит точка, которая правее всех корней, и точка, которая левее всех корней. Тогда формула

$$\mathcal{B} = \begin{cases} \bigvee\limits_{x_0 \in X} \Phi(x_0), & \text{если } Q = \exists \\ \bigwedge\limits_{x_0 \in X} \Phi(x_0), & \text{если } Q = \forall \end{cases}$$

свободна от вхождений кванторов и эквивалента формуле $\mathcal{A}$.

По теореме Ролля о нуле производной получаем, что для любой пары x_1, x_2 корней многочлена существует такая точка ξ, что ξ расположена между x_1, x_2, и производная многочлена в точке ξ обращается в ноль. То есть в качестве точек между корнями многочлена можно брать корни производной этого многочлена.

Множество корней многочлена

$$\prod_{i=1}^{n} f_i(x)$$

совпадает с объединением множеств корней многочленов $f_1(x), ..., f_n(x)$, тогда для набора многочленов в качестве точек между корнями удобно рассматривать корни многочлена

$$f_0(x) = \left(\prod_{i=1}^{n} f_i(x)\right)' .$$

1.4.1 Таблица Тарского

Пусть $f_1(x)...f_n(x)$ – многочлены, входящие в формулу $\Phi(x)$, $X = \{x_1, ..., x_s\}$ и $x_1 \leq ... \leq x_s$. Построим следующую таблицу:

	x_1	...	x_j	...	x_s
f_1	$f_1(x_1)$	...	$f_1(x_j)$	...	$f_1(x_s)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
f_i	$f_i(x_1)$	...	$f_i(x_j)$	...	$f_i(x_s)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
f_n	$f_n(x_1)$	...	$f_n(x_j)$	...	$f_n(x_s)$

Так как нас интересуют только знаки многочленов в этих точках, то вместо значений многочлена можно записывать его знак в этой точке:

	x_1	...	x_j	...	x_s
f_1	ε_{11}	...	ε_{1j}	...	ε_{1s}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
f_i	ε_{i1}	...	ε_{ij}	...	ε_{is}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
f_n	ε_{n1}	...	ε_{nj}	...	ε_{ns}

где

$$\varepsilon_{ij} = \begin{cases} +, & \text{если } f_i(x_j) > 0 \\ 0, & \text{если } f_i(x_j) = 0 \\ -, & \text{если } f_i(x_j) < 0 \end{cases} .$$

Таблицы такого вида будем называть **таблицами Тарского**.

Утверждение 1.2. *Знаки $+$ и $-$ не могут стоять в двух соседних по горизонтали клетках таблицы Тарского.*

Доказательство. Следует из теоремы Коши о нулях непрерывной функции, непрерывности многочленов как функций и упорядоченности точек $x_1, ..., x_n$. ◀

Нетрудно заметить, что имея таблицу Тарского, можно определить истинностное значение формулы $(Qx)\Phi(x)$, так как по таблице легко определяются истинностные значения атомарных формул, при этом уже нет необходимости знать сами точки $x_1, ..., x_n$. А зная истинностное значение формулы, можно построить эквивалентную бесквантороную:

- если формула истинна – то можно использовать любую тождественно истинную формулу, например, $0 = 0$;
- если формула ложна – то можно использовать любую тождественно ложную формулу, например $0 = 1$.

До сих пор не обсуждалось, как искать корни многочленов. Оказывается, таблицу Тарского можно построить не находя ни одного корня, если рассмотреть системы многочленов особого вида.

1.4.2 Насыщенная система

Определение 1.2. *[1] Система функций называется* ***полунасыщенной****, если вместе с каждой функцией она содержит и ее производную.*

Утверждение 1.3. *[1] Каждую конечную систему многочленов можно расширить до конечной полунасыщенной системы.*

Утверждение 1.4. *[1] Если многочлен отличен от тождественного нуля, то в строке таблицы Тарского, соответствующей этому многочлену, в соседних по горизонтали клетках не могут стоять два символа 0.*

Определение 1.3. *[1] Полунасыщенная система многочленов $p_1(x), ..., p_n(x)$ называется* ***насыщенной****, если вместе с каждыми двумя многочленами $p_i(x)$ и $p_j(x)$ такими, что $0 < deg(p_j(x)) \leq deg(p_i(x))$, она содержит и остаток $r(x)$ от деления $p_i(x)$ на $p_j(x)$.*

Утверждение 1.5. *[1] Каждую конечную систему многочленов можно расширить до конечной насыщенной системы.*

Утверждение 1.6. *[1] Если $p_1(x), ..., p_{n-1}(x), p_n(x)$ – насыщенная система многочленов, и*

$$deg(p_1(x)) \leq ... \leq deg(p_{n-1}(x)) \leq deg(p_n(x)),$$

то система $p_1(x), ..., p_{n-1}(x)$ также является насыщенной.

Утверждение 1.7. *Если $p_1(x), ..., p_n(x)$ – насыщенная система многочленов, и*

$$deg(p_1(x)) \leq deg(p_i(x)), \text{ где } i = 2, 3, ..., n,$$

то $deg(p_1(x)) < 1$.

Доказательство. Если предположить противное, то получим, что с одной стороны, в силу насыщенности, система должна содержать многочлен $p_1'(x)$, степень которого меньше $deg(p_1(x))$, а с другой стороны, степени всех многочленов должны быть не меньше $deg(p_1(x))$, противоречие. ◀

1.4.3 Метод построения таблицы Тарского

Пусть $p_1(x), ..., p_{n-1}(x), p_n(x)$ – насыщенная система многочленов, и многочлены упорядочены в ней по не убыванию степени.

Рассмотрим подсистему из одного элемента $p_1(x)$. Согласно утверждению 1.6, система $p_1(x)$ является насыщенной, тогда многочлен $p_1(x)$ представляет собой константу, так как его степень меньше единицы, согласно утверждению 1.7. В таком случае знак многочлена в любой точке совпадает со знаком этой константы. Таблица Тарского для одного многочлена имеет вид:

	$-\infty$	$+\infty$
p_1	ε	ε

Символами $-\infty$ и $+\infty$ обозначены точки, которые заведомо расположены левее и правее всех корней соответственно. При этом не надо выбирать конкретные значения для этих точек, так как в точке $+\infty$ знак многочлена совпадает со знаком старшего коэффициента, а знак в точке $-\infty$ зависит ещё и от степени многочлена:

- если степень многочлена четная – то знак в этой точке совпадает со знаком старшего коэффициента;
- иначе – равен знаку противоположному к знаку старшего коэффициента.

В таблице всего два столбца, поэтому для каждого столбца j, за исключением самого правого и самого левого, в этой таблице существует ненулевой многочлен $p_i(x)$ такой, что $\varepsilon_{ij} = 0$.

Индуктивное предположение: пусть для насыщенной системы $p_1(x), ..., p_{k-1}(x)$ уже построена таблица Тарского:

	$-\infty$	...		...	$+\infty$
p_1	$\varepsilon_{1,1}$	...	$\varepsilon_{1,j}$	...	$\varepsilon_{1,s}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
p_{k-1}	$\varepsilon_{k-1,1}$	...	$\varepsilon_{k-1,j}$	...	$\varepsilon_{k-1,s}$

И для каждого столбца j, за исключением самого правого и самого левого, в этой таблице существует ненулевой многочлен $p_i(x)$ такой, что $\varepsilon_{i,j} = 0$.

К этой таблице добавим строку для многочлена $p_k(x)$, записав знаки для крайних столбцов.

	$-\infty$	...		...	$+\infty$
p_1	$\varepsilon_{1,1}$	...	$\varepsilon_{1,j}$	...	$\varepsilon_{1,s}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
p_{k-1}	$\varepsilon_{k-1,1}$	...	$\varepsilon_{k-1,j}$	...	$\varepsilon_{k-1,s}$
p_k	$\varepsilon_{k,1}$	...		...	$\varepsilon_{k,s}$

Возьмем многочлен $p_i(x)$ такой, что $\varepsilon_{i,j} = 0$. Такой многочлен существует и отличен от тождественного нуля в силу индуктивного предположения.

Утверждение 1.8. *Пусть $f(x)$ и $g(x)$ – ненулевые многочлены. Если $g(a) = 0$, то $f(a) = r(a)$, где $r(x)$ – остаток от деления многочлена $f(x)$ на $g(x)$.*

Доказательство. Многочлен $r(x)$ – остаток от деления, тогда $f(x) = q(x)g(x)+r(x)$, подставив a получим $f(a) = q(a)g(a) + r(a) = q(a) \cdot 0 + r(a) = r(a)$. ◀

Найдём $p_t(x)$ – остаток от деления $p_k(x)$ на $p_i(x)$. Так как система многочленов насыщена, то многочлен $p_t(x)$ уже добавлен в таблицу. Тогда по только что доказанному утверждению, $\varepsilon_{k,j} = \varepsilon_{t,j}$. Таким образом, заполняется вся нижняя строка.

Просмотрим значения в нижней строке. Может случиться так, что в соседних клетках стоят знаки + и −. В таком случае необходимо добавить новый столбец между столбцами, в которых находятся эти клетки. Понятно, что в новом столбце нижняя клетка заполняется нулем, что следует из теоремы Коши о нулях непрерывной функции. Для оставшихся клеток рассмотрим случаи.

+	?	+

+	?	0

Тогда вместо символа ? ставится знак +.

0	?	+

0	?	−

В этих случаях ставится знак + или − соответственно.

−	?	0

−	?	−

А в этих – знак $-$.

При этом остальные случаи

+	?	−

−	?	+

0	?	0

невозможны по сформулированным ранее утверждениям о таблице Тарского.

Таким образом, удалось построить Таблицу Тарского для насыщенной системы многочленов $p_1(x), ..., p_{k-1}(x), p_k(x)$, при этом для каждого столбца найдется многочлен, на пересечении строки которого с выбранным столбцом в клетке записан символ 0.

1.4.4 Алгоритм для формулы вида $(Qx)\Phi(x)$

Все готово, чтобы описать алгоритм Тарского для формулы $\mathcal{A} = (Qx)\Phi(x)$:

1. Составить список всех многочленов $f_1(x), ..., f_n(x)$, входящих в $\Phi(x)$ и отличных от тождественного нуля;
2. Добавить к этому списку многочлен
$$f_0(x) = \left(\prod_{i=1}^{n} f_i(x)\right)';$$
3. Расширить этот список до насыщенной системы $p_1(x), ..., p_m(x)$, упорядоченной по не убыванию степени;
4. Построить таблицу Тарского, по очереди добавляя многочлены;
5. Вычислить истинностное значение $\Phi(x)$ для каждого из столбцов таблицы;
6. Если $Q = \exists$, то формула $\mathcal{A}$ истинна тогда и только тогда, когда хотя бы одно из вычисленных значений истинно. Если $Q = \forall$, то формула $\mathcal{A}$ истинна тогда и только тогда, когда все вычисленные значения истинны.

1.4.5 Алгоритм для формулы вида $(Qx)\Phi(x, a_1, ..., a_l)$

Оказывается, в случае, когда формула имеет вид $(Qx)\Phi(x, a_1, ..., a_l)$, нужно лишь немного модифицировать алгоритм. Во-первых, коэффициенты многочленов теперь не из $\mathbb{Q}$, а из поля частных целостного кольца $\mathbb{Q}[a_1, ..., a_l]$. Во-вторых, нельзя говорить о знаках таких коэффициентов, поэтому каждый раз, когда необходимо определить знак коэффициента, придется разбирать три случая: коэффициент меньше нуля, больше нуля или равен нулю. То есть будет построено дерево разбора случаев. Рано или поздно, разбор случаев закончится и будет построена таблица Тарского, и если по таблице получается, что формула истинна, тогда все предположения, сделанные в ходе разбора случаев, выписываются в виде конъюнкции. Результатом же работы алгоритма будет дизъюнкция всех таких конъюнкций.

1.4.6 Пример работы алгоритма

Пример из презентации

1.5 Теоремы Тарского

В результате можно сформулировать теорему:

Теорема 1.1 (Альфред Тарский)**.** *Для любой формулы $\mathcal{A}$ языка элементарной алгебры существует эквивалентная ей бесквантворная формула этого же языка.*

Оценки: количество многочленов, количество многочленов в насыщенной системе, размеры таблицы, количество ветвлений.

Стоит сказать, что алгоритм, предложенный Тарским в его работе [7], записывался иначе – предельно формально. Однако в литературе, например, в работе [5], именно описанный выше алгоритм носит имя Альфреда Тарского. Современное, менее формальное описание алгоритма доступнее для понимания, что, например, упрощает программисту его работу – реализацию алгоритма.

2 Программная реализация

Для реализации программы, которая по простой формуле (определение дать) языка элементарной алгебры, вводимой пользователем с клавиатуры, строит эквивалентную бесквантворную формулу, были выбраны язык C#, платформа .NET Core 3.1 и спецификация .NET Standard 2.1 [6]. Такой выбор обусловлен рядом причин:

- Язык C# – это объектно-ориентированный язык программирования, а данная парадигма программирования позволяет абстрактно описывать объекты, в том числе и математические объекты;
- .NET Core и .NET Standard – это современные, развивающиеся и востребованные кроссплатформенные технологии с открытым исходным кодом;
- Личные предпочтения автора.

Для поэтапного создания программы, были сформулированы и решены следующие задачи:

- Разработать библиотеку классов для объектов языка элементарной алгебры: символы алфавита, термы, формулы;
- Реализовать ввод логических формул;
- Разработать библиотеку классов для рациональных чисел и многочленов от одной вещественной переменной с рациональными коэффициентами;
- Реализовать алгоритм Тарского для базового случая;
- Реализовать алгоритм элиминации кванторов(ссылка на утверждение о сведении задачи к алгоритму тарского) для простых формул языка элементарной алгебры;
- Обеспечить минимальное покрытие юнит-тестами;
- Собрать все библиотеки и модули в единый программный комплекс.

Написание программы осуществлялось в среде разработки Microsoft Visual Studio 2019, доступной для использования в некоммерческих целях и скачивания на официальном сайте: https://visualstudio.microsoft.com/ru/.

Также была рассмотрена перспектива реализации алгоритма Тарского для общего случая.

2.1 Представление формул

форма представления символов и формул языка
Ввести понятие нуль-местного предиката.

2.2 Система ввода

Ввод формул

2.3 Реализация базового случая

Часть математики, часть алгоритм. Runner. Покрытие тестами. Консольное приложение

SimpleTarskiAlgorithmRunner

2.4 Юнит-тестирование

тесты

2.5 Результаты работы

Демонстрация результатов

2.6 Общий случай

Часть математики. Описание нерешённых задач.

Заключение

Заключение

Список литературы

[1] Алгоритм Тарского. Лекция 1 // Лекториум. URL: https://www.lektorium.tv/lecture/31079 (дата обращения: 01.12.2019).

[2] Алгоритм Тарского. Лекция 2 // Лекториум. URL: https://www.lektorium.tv/lecture/31080 (дата обращения: 01.12.2019).

[3] Гибадулин Р. А. Алгоритм поиска вывода в Исчислении Высказываний и его программная реализация // Современные проблемы математики и информатики : сборник научных трудов молодых ученых, аспирантов и студентов. / Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль : ЯрГУ, 2019. – Вып. 19. – С. 28-37.

[4] Дурнев, В. Г. Элементы теории множеств и математической логики: учеб. пособие / Яросл. гос. ун-т. им. П. Г. Демидова, Ярославль, 2009 – 412 с.

[5] Матиясевич, Ю. В. Алгоритм Тарского // Компьютерные инструменты в образовании. – 2008. – № 6. – С. 14.

[6] Троелсен, Э. Язык программирования C# 7 и платформы .NET и .NET Core / Э. Троелсен, Ф. Джепикс; пер. с англ. и ред. Ю.Н. Артеменко. – 8-е изд. – М.; СПб.: Диалектика, 2020. – 1328 с.

[7] Tarski, A. A Decision Method for Elementary Algebra and Geometry: Prepared for Publication with the Assistance of J.C.C. McKinsey, Santa Monica, Calif.: RAND Corporation, R-109, 1951.

Приложение А

Приложение