Матиясевич Юрий Владимирович

АЛГОРИТМ ТАРСКОГО

Аннотация

Алгоритм Тарского позволяет установить истинность или ложность любого утверждения про конечное количество вещественных чисел. Вместе с методом координат Декарта это позволяет автоматически доказывать широкий класс теорем элементарной геометрии.

Изложенный здесь вариант алгоритма предназначен для первоначального знакомства с этой областью – его нетрудно понять, несложно запрограммировать, но полученная программа будет крайне неэффективной.

Ключевые слова: алгоритм Тарского, разрешимость элементарной алгебры и геометрии, элиминация кванторов.

Алгебра – это арифметика для лентяев.
К. Гаусс

По аналогии с высказыванием великого немецкого математика Гаусса, эту статью можно было бы назвать «Алгебра – это геометрия для лентяев». Первый шаг к обоснованию этого тезиса сделал Рёне Декарт, введя свой метод координат. Рассмотрим этот шаг подробнее на примере планиметрии.

В геометрии традиционно рассматриваются различные объекты: *точки* – неопределяемые понятия, *прямые*, состоящие из точек, *окружности*, также состоящие из точек, и т. д. Между этими объектами вводятся первичные отношения – *точка A лежит на прямой l*, *точка A лежит на окружности O* и т. д.

Нетрудно, однако, увидеть, что значительную часть планиметрии можно изложить, говоря только о точках и отношениях между ними. Вместо прямых достаточно иметь трехместное отношение OnLine(A, B, C), говорящее что *точки A, B и C лежат на одной прямой*, окужности легко заменяются четырёхместным отношением EqDistance(A, B, C, D), говорящим что *расстояние между точками A и B равно расстоянию между точками C и D* и т. д. Вот пример теоремы, записанной таким языком (см. рис. 1).

Узнали эту теорему? Здесь всего навсего сказано, что три медианы A_1B_1, A_2B_2 и A_3B_3 в треугольнике $\Delta A_1A_2A_3$ пересекаются в одной точке – точке C.

Мы можем заменить каждую точку A парой вещественных чисел (a_x, a_y) – её координатами, а вместо отношений между точками использовать соответствующие отношения между числами: OnLine(a_x, a_y, b_x, b_y, c_x, c_y) – *точки с координатами* (a_x, a_y), (b_x, b_y) *и* (c_x, c_y) *лежат на одной прямой,* EqDistance(a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y) – *расстояние между точками с координатами* (a_x, a_y) *и* (b_x, b_y) *равно расстоянию между точками с координатами* (c_x, c_y) *и* (d_x, d_y). Эти отношения легко записываются алгебраическими уравнениями:

Теорема. Каковы бы ни были точки A_1, A_2 и A_3, существуют точки B_1, B_2, B_3 и C такие, что

$$A_1 \neq A_2 \;\&\; A_1 \neq A_3 \;\&\; A_2 \neq A_3 \Rightarrow$$
$$\Rightarrow \mathrm{OnLine}(A_1, A_2, B_3) \;\&\; \mathrm{EqDistance}(A_1, B_2, B_2, A_3) \;\&$$
$$\&\; \mathrm{OnLine}(A_2, A_3, B_1) \;\&\; \mathrm{EqDistance}(A_2, B_1, B_1, A_3) \;\&$$
$$\&\; \mathrm{OnLine}(A_1, A_3, B_2) \;\&\; \mathrm{EqDistance}(A_1, B_3, B_3, A_2) \;\&$$
$$\&\; \mathrm{OnLine}(A_1, B_1, C) \;\&\; \mathrm{OnLine}(A_2, B_2, C) \;\&\; \mathrm{OnLine}(A_3, B_3, C).$$

Рис. 1

$$\mathrm{OnLine}(a_x, a_y, b_x, b_y, c_x, c_y) \Leftrightarrow$$
$$\Leftrightarrow a_x b_y + a_y c_x + b_x c_y - a_x c_y - a_y b_x - b_y c_x = 0$$

$$\mathrm{EqDistance}(a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y) \Leftrightarrow$$
$$\Leftrightarrow (a_x - b_x)^2 + (a_y - b_y)^2 = (c_x - d_x)^2 + (c_y - d_y)^2$$

В результате наша теорема приобретает вид следующего утверждения про числа, (см. рис. 2).

Что же нам делать дальше с таким ужасным выражением, в которое превратилась знакомая теорема о пересечении медиан? Вот тут нам и придёт на помощь алгоритм Тарского. Этот алгоритм позволяет установить истинность или ложность любого замкнутого утверждения про вещественные числа. Прежде чем описывать алгоритм, давайте четко опишем класс утверждений, к которым можно применять алгоритм Тарского.

Первым делом мы введем формальный язык $\mathcal{A}$ (язык алгебры).

Язык $\mathcal{A}$ содержит:

- *обозначения* для всех рациональных чисел;
- *переменные* для вещественных чисел;
- *знаки операций* сложения и умножения;
- *знаки отношений* $=$, $>$, $<$;
- *логические связки* $\&$, $\vee$, $\neg$, $\Rightarrow$;
- *кванторы* $\forall$, $\exists$.

В качестве обозначений для рациональных чисел можно использовать, например, обыкновенные дроби – 2, –3, 5/7, –451/53, ... и конечные десятичные дроби – 3,14159265, 2,71828, ...; мы, однако, не можем иметь в нашем языке обозначения для всех вещественных чисел – бесконечная десятичная дробь не может быть подана на вход никакого алгоритма.

Теорема. Каковы бы ни были числа $a_{1,x}$, $a_{1,y}$, $a_{2,x}$, $a_{2,y}$, $a_{3,x}$, $a_{3,y}$, существуют числа $b_{1,x}$, $b_{1,y}$, $b_{2,x}$, $b_{2,y}$, $b_{3,x}$, $b_{3,y}$ и c_x, c_y такие, что

$$(a_{1,x} \neq a_{2,x} \vee a_{1,y} \neq a_{2,y}) \;\&\; (a_{1,x} \neq a_{3,x} \vee a_{1,y} \neq a_{3,y}) \;\&\; (a_{2,x} \neq a_{3,x} \vee a_{2,y} \neq a_{3,y}) \Rightarrow$$
$$\Rightarrow a_{1,x}a_{2,y} + a_{1,y}b_{3,x} + a_{2,x}b_{3,y} - a_{1,x}b_{3,y} - a_{1,y}a_{2,x} - a_{2,y}b_{3,} = 0 \;\&$$
$$\&\; (a_{1,x} - b_{2,x})^2 + (a_{1,y} - b_{2,y})^2 = (b_{2,x} - a_{3,x})^2 + (b_{2,y} - a_{3,y})^2 \;\&$$
$$\&\; a_{2,x}a_{3,y} + a_{2,y}b_{1,x} + a_{3,x}b_{1,y} - a_{2,x}b_{1,y} - a_{2,y}a_{3,x} - a_{3,y}b_{1,x} = 0 \;\&$$
$$\&\; (a_{2,x} - b_{1,x})^2 + (a_{2,y} - b_{1,y})^2 = (b_{1,x} - a_{3,x})^2 + (b_{1,y} - a_{3,y})^2 \;\&$$
$$\&\; a_{1,x}a_{3,y} + a_{1,y}b_{2,x} + a_{3,x}b_{2,y} - a_{1,x}b_{2,y} - a_{1,y}a_{3,x} - a_{3,y}b_{2,x} = 0 \;\&$$
$$\&\; (a_{1,x} - b_{3,x})^2 + (a_{1,y} - b_{3,y})^2 = (b_{3,x} - a_{2,x})^2 + (b_{3,y} - a_{2,y})^2 \;\&$$
$$\&\; a_{1,x}b_{1,y} + a_{1,y}c_x + b_{1,x}c_y - a_{1,x}c_y - a_{1,y}b_{1,x} - b_{1,y}c_x = 0 \;\&$$
$$\&\; a_{2,x}b_{2,y} + a_{2,y}c_x + b_{2,x}c_y - a_{2,x}c_y - a_{2,y}b_{2,x} - b_{2,y}c_x = 0 \;\&$$
$$\&\; a_{3,x}b_{3,y} + a_{3,y}c_x + b_{3,x}c_y - a_{3,x}c_y - a_{3,y}b_{3,x} - b_{3,y}c_x = 0.$$

Рис. 2

Таким образом, хотя мы собираемся узнавать истинность или ложность утверждений про *вещественные* числа, в самих утверждениях могут встречаться только конкретные *рациональные* числа.

В качестве переменных для вещественных чисел мы будем использовать строчные латинские буквы с индексами и без них – a, b, c, ..., a_1, b_2, x_6, ...

Для успеха алгоритма очень важно, что у нас имеются только переменные, допустимыми значениями которых являются все вещественные числа – для аналогичного языка с переменными, принимающими только рациональные значения, алгоритм, подобный алгоритму Тарского, невозможен.

Имея в своем распоряжении обозначения для рациональных чисел, переменные для вещественных чисел и знаки операций сложения и умножения, мы можем строить по обычным правилам более сложные выражения – *многочлены*. Мы можем включить в язык скобки для указания порядка применения операций, что позволит сократить длину записей, но никак не увеличит выразительную силу языка – скобки всегда можно раскрыть.

С помощью знаков отношений мы можем строить из многочленов *элементарные формулы*, например,

$$x^2y + 4xy^3 > (x - y)^2, \qquad (1)$$

$$xy = 3x + 2y. \qquad (2)$$

С помощью логических связок & («и»), $\vee$ («или»), $\neg$ («не») и $\Rightarrow$ («если ..., то ...») из элементарных формул можно строить более сложные *формулы* по следующим правилам: *если* Φ *и* Ψ *– формулы, то* $(\Phi\ \&\Psi)$, $(\Phi \vee \Psi)$, $\neg\Phi$, $(\Phi \Rightarrow \Psi)$ *также являются формулами.*

Из элементарных формул (1) и (2) мы можем образовать, например, формулу

$$x^2y + 4xy^3 > (x\ \ - y)^2\ \&\ xy = 3x + 2y. \qquad (3)$$

Мы, однако, не можем задавать вопрос, верна ли формула (3) сама по себе, поскольку она содержит переменные. Однако можно спросить: *Верно ли* (3) *при* $x = 4$, $y = 5$*?*

Существует и другой путь сделать осмысленным вопрос об истинности или ложности (3), а именно, можно спросить следующее:

- *Верно ли* (3) *при любых* x, y*?*
- *Существуют ли* x, y *такие, что выполнено* (3)*?*
- *Верно ли, что для любого* x *существует* y *такое, что выполнено* (3)*?*

В нашем языке $\mathcal{A}$ для постановки таких вопросов имеются кванторы: *квантор общности* $\forall$ («для всех») и квантор существования $\exists$ («существует»).

С помощью кванторов из уже построенных формул можно строить ещё более сложные по следующим правилам: *если* Φ *– формула, а* α *– переменная, то* $\forall\alpha\{\Phi\}$ *и* $\exists\alpha\{\Phi\}$ *также являются формулами}*. При этом говорят, что квантор *связывает* стоящую за ним переменную.

Приведённые выше вопросы – это вопросы об истинности следующих формул нашего языка:

$$\forall x\{\forall y\{(x^2y + 4xy^3 > (x - y)^2\ \&\ xy = 3x + 2y)\}\},$$
$$\exists x\{\exists y\{(x^2y + 4xy^3 > (x - y)^2\ \&\ xy = 3x + 2y)\}\},$$
$$\forall x\{\exists y\{(x^2y + 4xy^3 > (x - y)^2\ \&\ xy = 3x + 2y)\}\}, \qquad (4)$$

Рассмотрим ещё формулу

$$\forall x\{(\{\exists y\{x^2y + 4xy^3 > (x - y)^2\}\ \&$$
$$\&\ \exists y\{xy = 3x + 2y\})\}, \qquad (5)$$

На первый взгляд она эквивалентна формуле (4) – и там, и там переменная x связана квантором общности, а переменная y – квантором существования. На самом деле формула (5) слабее формулы (4). Дело в том, что у каждого квантора в любой формуле есть своя *область действия*, заканчивающаяся закрывающей скобкой, парной к открывающей скобке, стоящей сразу за связываемой переменной. В формуле (4) утверждается существование одного числа y, удовлетворяющего одновременно и (1), и (2), в то время как в формуле (5) утверждается существование двух, вообще говоря разных чисел, одно из которых удовлетворяет (1), а другое – (2).

По аналогичной причине квантор существования в формуле

$$\forall x\{(\{\exists y\{x^2y + 4xy^3 > (x - y)^2\}\ \&\ xy = 3x + 2y)\},$$

не связывает переменную из второго члена конъюнкции, и поэтому нельзя спрашивать, верна ли эта формула.

Формула, в которой все вхождения переменных связаны кванторами, называется *замкнутой*. Именно про замкнутые формулы можно спрашивать, верны ли они или нет, и алгоритм Тарского позволяет находить ответ.

Вот пример задачи, к которой не понятно как подступиться методами «школьной» геометрии, но которая легко переводится на язык алгебры.

Шесть одинаковых монет можно расположить на столе так, чтобы они не перекрывали друг друга, но касались бы седьмой такой же монеты. А сколько бильярдных шаров могут касаться одного шара? Легко видеть, что по крайней мере двенадцать: заменим семь монет на семь шаров и добавим ещё три сверху и три снизу. А нельзя ли, пошевелив шары, уместить и тринадцатый шар?

Это было предметом спора между Исааком Ньютоном и Давидом Грегори, первый считал, что 13 шаров разместить нельзя, второй – что можно. Сегодня вы легко можете записать эту проблему на нашем языке $\mathcal{A}$, после чего для получения ответа останется применить алгоритм Тарского.

Алгоритм Тарского работает по индукции по числу кванторов в формуле. Случай, когда кванторов нет вообще, тривиален – в этом случае не может быть и переменных, и мы можем сначала вычислить значения всех встречающихся в формуле многочленов, а потом найти истинностные значения – ИСТИНА или ЛОЖЬ – всех элементарных подформул, а затем и всей формулы в целом.

Рассмотрим теперь базисный случай замкнутой формулы $Qx\{\Phi(x)\}$ с одним квантором, где Q – либо квантор существования $\exists$, либо квантор общности $\forall$. Основная идея алгоритма Тарского проявляется уже в случае, когда $\Phi(x)$ – элементарная формула.

Рассмотрим сначала пример, когда $\Phi(x)$ является формулой $P(x) = 0$, где

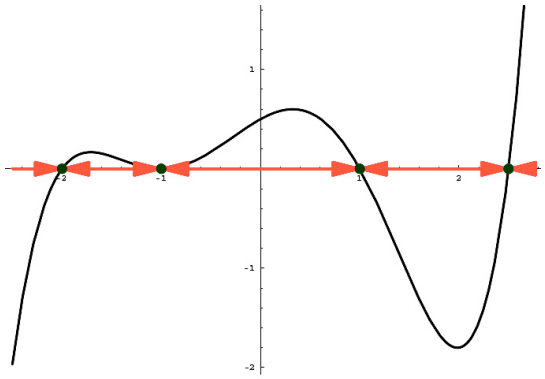

Рис. 3

$$P(x)=\frac{1}{2}+\frac{11}{20}x-\frac{11}{20}x^2-\frac{13}{20}x^3+$$
$$+\frac{1}{20}x^4+\frac{1}{10}x^5.$$

Мы можем отметить на вещественной оси те значения x, при которых формула истинна – это нули многочлена $P(x)$ (см. рис. 3), их количество конечно, оно не превосходит степени многочлена. После удаления этих точек остаётся конечное количество открытых промежутков, на которых формула ложна. Концы этих промежутков – это нули многочлена $P(x)$ и, возможно, точки $-\infty$ и $+\infty$. Для наших целей все точки каждого из этих промежутков как бы «на одно лицо», нам важно только то, что количество таких промежутков конечно.

Аналогично формула $P(x) > 0$ истинна на конечном количестве открытых промежутков, концы которых – нули многочлена и, возможно, точки $-\infty$ и $+\infty$ (см. рис. 4), и

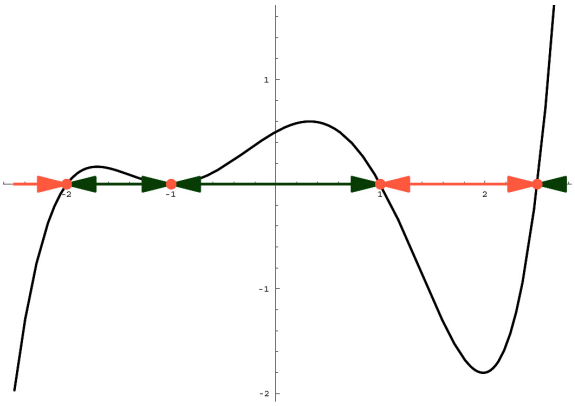

Рис. 4

ложна на конечном количестве дополнительных промежутков. Опять таки для наших целей все точки каждого из этих промежутков «неразличимы», и нам важно только то, что количество таких промежутков конечно.

Теперь можно описать «алгоритм» Тарского для случая одноквантерной формулы (см. рис. 5). Слово алгоритм взято здесь в кавычки, поскольку пока не ясно, как выполнить некоторые шаги.

Не ограничивая общности, мы будем предполагать, что правые части всех элементарных формул – это число 0 (ибо всегда можно перенести все члены равенства или неравенства в левую часть).

Как мы видим, хотя в формуле $Qx\{\Phi(x)\}$ стоит квантор по бесконечному множеству вещественных чисел, для установления истинности или ложности этой формулы нам достаточно узнать истинностные значения формулы $\Phi(x)$ на конечном множестве специально отобранных чисел – достаточно, чтобы среди них были корни всех многочленов, встречающихся в $\Phi(x)$, и хотя бы по одной точке из каждого промежутка, возникающего при удалении всех этих корней.

Чтобы превратить описанный выше «алгоритм» в настоящий алгоритм, надо уточнить, что значит *найти все корни* и *выбрать по числу между ними*. Мы начнём с уточнения второго действия.

По теореме Ролля из курса математического анализа мы знаем, что производная дифференцируемой функции обращается в ноль в некоторой точке между любыми двумя различными нулями функции. Мы можем использовать эту теорему, чтобы зафиксировать способ выбора дополнительных точек (см. рис. 6).

Оформим работу нашего алгоритма в виде *таблицы Тарского*. Строки такой таблицы будут помечены некоторыми многочленами $T_1(x)$, ..., $T_m(x)$, степени которых $\deg(T_1(x))$, ..., $\deg(T_m(x))$ идут в порядке неубывания: $\deg(T_1(x)) \le \ldots \le \deg(T_m(x))$. Столбцы таблицы, кроме двух крайних, помечены некоторыми числами x_1, ..., x_n, идущими в порядке возрастания; крайне левый и крайне правый столбцы помечены, соответственно, символами $-\infty$ и $+\infty$. В клетке, находящейся на пересечении строки, помеченной многочленом $T_i(x)$, и столбца, помеченного числом x_j, будет находиться один из трёх символов –, 0 или +, в соответствии с тем, какое из трех условий $T_i(x_j) < 0$, $T_i(x_j) = 0$ или же $T_i(x_j) > 0$ выполнено. Аналогично, содержимое клетки, находящейся на пересечении строки, помеченной многочленом $T_i(x)$, и столбца, помеченного символом $-\infty$ или $+\infty$, бу-

«Алгоритм» Тарского (1-я версия) для $Qx\{\Phi(x)\}$

1. Составить список $P_1(x)$, ..., $P_k(x)$ всех многочленов, входящих в $\Phi(x)$ и отличных от тождественного нуля.

2. Найти множество $\mathfrak{N} = \{x_0, \ldots, x_n\}$, состоящее из всех корней всех многочленов $P_1(x)$, ..., $P_k(x)$; не ограничивая общности, мы считаем, что

$$x_0 < x_1 < \ldots < x_{n-1} < x_n$$

3. Расширить множество $\mathfrak{N}$ до множества $\mathfrak{M} = \{y_0, \ldots, y_m\} \supset \mathfrak{N}$ такого, что
- для любого i, такого что $0 < i \le n$, существует j, такое что $0 < j \le n$ и $x_{i-1} < y_j < x_i$;
- для любого i, такого что $0 < i \le n$, $y_0 < x_i$;
- для любого i, такого что $0 < i \le n$, $x_i < y_m$;

4. Формула $\exists x\{\Phi(x)\}$ истинна, если и только если $\Phi(y_0) \vee \ldots \vee \Phi(y_m)$
Формула $\forall x\{\Phi(x)\}$ истинна, если и только если $\Phi(y_0)$ & ... & $\Phi(y_m)$

Рис. 5

«Алгоритм» Тарского (2-я версия) для $Qx\{\Phi(x)\}$

1. Составить список $P_1(x)$, ..., $P_k(x)$ всех многочленов, входящих в $\Phi(x)$ и отличных от тождественного нуля.

2. Добавить многочлен $P_0(x) = (P_1(x) \cdot ... \cdot P_k(x))'$.

3. Найти множество $\mathfrak{N} = \{x_0, ..., x_n\}$, состоящее из всех корней всех многочленов $P_0(x)$, $P_1(x)$, ..., $P_k(x)$.

4. Расширить множество $\mathfrak{N}$ до множества $\mathfrak{M} = \{x_{-\infty}, x_0, x_1, ..., x_{+\infty}\}$, где $x_{-\infty}$ и $x_{+\infty}$ – такие числа, что $x_{-\infty} < x_0 < x_1 < ... < x_{n-1} < x_n < x_{+\infty}$.

5. Формула $\exists x\{\Phi(x)\}$ истинна, если и только если $\Phi(x_{-\infty}) \vee \Phi(x_0) \vee ... \vee \Phi(x_n) \vee \Phi(x_{+\infty})$.
Формула $\forall x\{\Phi(x)\}$ истинна, если и только если $\Phi(x_{-\infty})$ & $\Phi(x_0)$ & ... $\vee$& $\Phi(x_n)$ & $\Phi(x_{+\infty})$.

Рис. 6

дет определяться тем, какое из трех условий $T_i(x) < 0$, $T_i(x) = 0$ или же $T_i(x) > 0$ выполнено для всех достаточно больших по абсолютной величине отрицательных или, соответственно, положительных значений x.

Таким образом, таблица Тарского выглядит так (см. табл. 1).

От каждой таблицы Тарского мы будем требовать выполнения двух следующих условий:

• Если некоторая строка помечена многочленом, который отличен от тождественно нулевого многочлена, и y – корень этого многочлена, то один из столбцов помечен числом y.

• Если некоторый столбец помечен числом y, то одна из строк помечена многочленом, который отличен от тождественно нулевого многочлена и для которого y является корнем.

Из этих свойств следует, в частности, что знаки – и + не могут стоять в двух соседних по горизонтали клетках – действительно, многочлен должен обратиться в нуль между двумя точками, где он принимает разные знаки, а этим нулем должен быть помечен один из столбцов.

Перейдем к рассмотрению 3-ей версии «алгоритма» Тарского (см. рис. 7).

Давайте будем ленивыми и не будем делать лишней работы. В 3-ей версии алгоритма при вычислении истинностных значений элементарных формул мы пользуется только содержимым таблицы, при этом метки стобцов не используются. Мы можем заменить полноценную таблицу Тарского на *сокращенную*, которая получается удалением всех меток столбцов, кроме двух крайних (можно сказать и наоборот: сокращенная таблица Тарского – это такая таблица, из которой можно получить стандартную таблицу путем добавления меток столбцов). С одной стороны, сокращенная таблица – это вполне конструктивный конечный объект, в

Табл. 1

	$-\infty$	x_0	...	x_j	...	x_n	$+\infty$
$T_1(x)$	–|0|+	–|0|+		–|0|+		–|0|+	–|0|+
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
$T_i(x)$	–|0|+	–|0|+	...	–|0|+	...	–|0|+	–|0|+
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
$T_l(x)$	–|0|+	–|0|+	...	–|0|+	...	–|0|+	–|0|+

«Алгоритм» Тарского (3-я версия) для $Qx\{\Phi(x)\}$

1. Составить список $P_1(x)$, ..., $P_k(x)$ всех многочленов, входящих в $\Phi(x)$ и отличных от тождественного нуля.

2. Добавить многочлен $P_0(x) = (P_1(x) \cdot \ldots \cdot P_k(x))'$.

3. Построить таблицу Тарского для многочленов $P_0(x)$, $P_1(x)$, ..., $P_k(x)$.

4. Вычислить логичекие значения $\Phi(x_j)$ для каждого столбца таблицы, пользуясь только содержимым таблицы:

	$-\infty$	x_0	...	x_j	...	x_n	$+\infty$
$P_0(x)$	–|0|+	–|0|+		–|0|+		–|0|+	–|0|+
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
$P_i(x)$	–|0|+	–|0|+	...	–|0|+	...	–|0|+	–|0|+
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
$P_k(x)$	–|0|+	–|0|+	...	–|0|+	...	–|0|+	–|0|+
$\Phi(x)$	И/Л	И/Л	...	И/Л	...	И/Л	И/Л

5. Формула $\exists x\{\Phi(x)\}$ истинна, если и только если хотя бы одно из этих значений истинно; формула $\forall x\{\Phi(x)\}$ истинна, если и только если все эти значения истинны.

Рис. 7

отличие от стандартной, содержащей вещественные числа. С другой стороны, оказывается, что построить сокращенную таблицу можно и не зная потенциальных меток столбцов.

Мы будем строить сокращенную таблицу Тарского индукцией по количеству многочленов, но будем делать это не для произвольных многочленов, а системы, организованной специальным образом.

Будем говорить, что система функций является *полунасыщенной*, если вместе с каждой функцией она содержит и ее производную. Легко видеть, что *каждую конечную систему многочленов можно расширить до конечной полунасыщенной системы*. Это следует из того, что производная многочлена, отличного от тождественного нуля, является многочленом меньшей степени, а производной тождественного нуля (формально являющегося многочленом степени $-\infty$) является он сам. По этой причине процесс расширения системы многочленов их производными закончится через конечное число шагов.

Легко видеть, что таблица Тарского, построенная для полунасыщенной системы многочленов, обладает следующим свойством: *если строка помечена многочленом, отличным от тождественного нуля, то в ней символ 0 не может стоять в двух соседних клетках*. Действительно, опять-таки по теореме Ролля производная многочлена должна обратиться в ноль между двумя нулями многочлена, и, следовательно, должен иметься столбец, помеченный точкой, где это происходит.

Далее, полунасыщенная система многочленов называется *насыщенной*, если для любых двух её многочленов $T_i(x)$ и $T_k(x)$, таких что $0 < \deg T_i(x) \le \deg T_k(x)$, в систему входит и остаток от деления $T_k(x)$ на $T_i(x)$, то есть многочлен $R(x)$ такой, что его степень строго меньше степени многочлена $T_i(x)$ и существует многочлен $S(x)$ такой, что

$$T_k(x) = S(x)\,T_i(x) + R(x). \qquad (6)$$

Остаток от деления многочлена на многочлен легко находится с помощью

«деления столбиком», аналогичного изучаемому в школе методу деления целых чисел.

Легко видеть, что *каждую конечную систему многочленов можно расширить до конечной насыщенной системы*. Это следует из того, что по определению степень остатка меньше степени делителя, и, следовательно, процесс расширения системы многочленов их производными и остатками от деления закончится через конечное число шагов.

Мы будем строить сокращенную таблицу Тарского для насыщенной системы многочленов $T_1(x)$, ..., $T_m(x)$, упорядоченной по невозрастанию степеней. Легко понять, что и каждый начальный отрезок $T_1(x)$, ..., $T_k(x)$, где $k \le m$, также будет насыщенной системой.

Начало построения тривиально. Если первые k многочленов являются константами, тогда сокращённая таблица Тарского для них имеет только два столбца, помеченных символами $-\infty$ и $+\infty$, и эти столбцы легко заполняются.

Пусть мы уже построили сокращённую таблицу Тарского для $T_1(x)$, ..., $T_{k-1}(x)$ и хотим её раширить до сокращённой таблицы Тарского для $T_1(x)$, ..., $T_k(x)$. Первым делом мы добавим снизу строку, которую пометим многочленом $T_k(x)$. Заполнить самую правую клетку новой строки легко – достаточно скопировать туда знак ведущего коэффициента этого многочлена. Заполнение самой левой клетки не намного сложнее: если многочлен $T_k(x)$ имеет чётную степень, то мы по-прежнему копируем знак ведущего коэффициента, если нечетную – то записываем противоположный знак (см. табл. 2).

Но как заполнить клетку, стоящую не в крайнем столбце? Вспомним, что он может быть помечен некоторым не известным нам корнем x_j какого-то многочлена $T_i(x)$. Поскольку система насыщена, она содержит и остаток $R(x)$ от деления $T_k(x)$ на $T_i(x)$, то есть выполнено (6) для некоторого многочлена $S(x)$. Подсталяя в (6) x_j на место x, мы видим, что $T_k(x_j) = R(x_j)$, так что нам достаточно скопировать в последнюю клетку столбца содержимое клетки, находящейся в строке, помеченной многочленом $R(x)$.

Таким образом, даже не имея меток столбцов, мы в состоянии заполнить всю нижнюю строку! Это, однако, ещё не всё, нам, быть может, требуется добавить новые столбцы, соответствующие корням многочлена $T_k(x)$. Где же искать место для этих добавочных столбцов?

Предположим, что в нижней строке в соседних клетках оказались знаки – и +. В этом случае мы кричим «Эврика!» и добавляем новый столбец, разделяя эти клетки. В нижнюю клетку нового столбца мы вписываем 0, но как заполнить остальные клетки нового столбца?

Предположим, что слева от некоторой пустой пока клетки стоит клетка со знаком +. Легко понять, что тогда и в эту пока пустую клетку мы должны также поместить знак +, поскольку корни всех «старых» многочленов уже были представлены в таблице. Аналогично, если слева от пустой клетки стоит знак –, мы копируем его. Если же слева от пустой клетки находится клетка со знаком 0, то мы можем посмотреть на клетку, соседнюю справа. Если там знак – или +, то мы копируем его, но что делать, если в обеих соседних клетках стоят нули? Как было отмечено ранее, такое возможно, лишь если эта строка была помечена тождественно нулевым многочленом, и мы, соответственно, вписываем также 0.

Табл. 2

	$-\infty$	...	x_j	...	$+\infty$
$T_1(x)$	–|0|+		–|0|+		–|0|+
⋮	⋮	⋱	⋮	⋱	⋮
$R(x)$	–|0|+	...	–|0|+	...	–|0|+
⋮	⋮	⋱	⋮	⋱	⋮
$T_i(x)$	–|0|+	...	–|0|+	...	–|0|+
⋮	⋮	⋱	⋮	⋱	⋮
$T_{k-1}(x)$	–|0|+	...	–|0|+	...	–|0|+
$T_k(x)$	–|0|+	...	?	...	–|0|+

Действуя подобным образом, мы можем заполнить все новые столбцы, но завершится ли на этом построение новой сокращённой таблицы Тарского? Можем ли мы быть уверены, что все корни многочлена $T_k(x)$ представлены столбцами?

Корни многочлена $T_k(x)$ могут быть простыми или кратными. Кратный корень является также корнем производной $T_k(x)'$ и потому уже представлен каким-то столбцом. Пусть y – простой корень, то есть $T_k(y) = 0$, но $T_k(y)' \neq 0$. Рассмотрим случай $T_k(y)' > 0$ (случай $T_k(y)' < 0$ аналогичен). Если y не является корнем ни одного из многочленов $T_1(x)$, ..., $T_{k-1}(x)$, то пусть $\check{x}$ – максимум тех из этих корней, которые не превосходят y (если таких корней нет, то по определению этот максимум есть $-\infty$). Аналогично пусть $\hat{x}$ – минимум тех корней, которые превосходят y (если таких корней нет, то по определению этот минимум есть $+\infty$). Нетрудно понять, что $T_k(x)' > 0$ при $x \in (\check{x}, \hat{x})$, столбцы, соответствующие $\check{x}$ и $\hat{x}$, являются соседними, и их нижние клетки содержат, соответственно, – и +. Таким образом, описанная выше процедура действительно выявит все требуемые новые столбцы.

Теперь мы наконец можем убрать кавычки вслове «алгоритм» (см. рис. 8).

Итак, алгоритм описан для базисного случая – замкнутой формулы с одним квантором. Прежде чем перейти к замкнутым формулам с произвольным количеством кванторов, мы рассмотрим формулы, имеющие лишь один квантор, но более одной переменной. Про такие формулы нельзя спрашивать, истинны они или ложны, с ними надо работать как с уравнениями или неравенствами с параметрами путём разбора случаев. К чему же мы можем стремиться?

В качестве примера рассмотрим формулу

$$\exists x\{ax^2 + bx + c = 0\}. \qquad (7)$$

В ней три параметра, при одном выборе значений параметров формула истинна, при других значениях – ложна. Оказывается, можно найти бескванторную формулу с теми же параметрами, которая

Алгоритм Тарского для $Qx\{\Phi(x)\}$

1. Составить список $P_1(x)$, ..., $P_k(x)$ всех многочленов, входящих в $\Phi(x)$ и отличных от тождественного нуля.

2. Добавить многочлен $P_0(x) = (P_1(x) \cdot \ldots \cdot P_k(x))'$.

3. Расширить этот список до насыщенной системы многочленов $T_1(x)$, ..., $T_m(x)$ с

$$\deg(T_1(x)) \le \ldots \le \deg(T_{i-1}(x)) \le \deg(T_i(x)) \le \ldots \le \deg(T_m(x)).$$

4. Последовательно построить сокращенные таблицы Тарского для многочленов $T_1(x)$, $T_2(x)$, ..., $T_i(x)$, $i = 0, 1, 2, \ldots, m$.

5. Вычислить логичекие значения $\Phi(x_j)$ для каждого столбца последней таблицы:

	$-\infty$						$+\infty$
$T_1(x)$	–|0|+	–|0|+		–|0|+		–|0|+	–|0|+
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
$T_m(x)$	–|0|+	–|0|+	...	–|0|+	...	–|0|+	–|0|+
$\Phi(x)$	И/Л	И/Л	...	И/Л	...	И/Л	И/Л

6. Формула $\exists x\{\Phi(x)\}$ истинна, если и только если хотя бы одно из этих значений истинно; формула $\forall x\{\Phi(x)\}$ истинна, если и только если все эти значения истинны.

Рис. 8

истинна или ложна одновременно с (7):

$$((a \neq 0 \ \& \ b^2 - 4ac \geq 0) \vee$$
$$\vee (a = 0 \ \& \ (b \neq 0 \vee c = 0))). \qquad (8)$$

Переход от (7) к (8) называется *элиминацией (устранением)* квантора.

Покажем, что в нашем языке кванторы можно элиминировать из любой формулы.

Пусть имеется формула вида

$$Qx\{\Phi(a_1, \ldots, a_m, x)\} \qquad (9)$$

с параметрами $a_1, \ldots, a_m$. На многочлены, входящие в эту формулу, можно смотреть как на многочлены от одной переменной x, коэффициенты которых являются многочленами от параметров. Нам потребуются более общие объекты – многочлены от одной переменной x, коэффициенты которых являются рациональными функциями от параметров, то есть отношениями пары многочленов. Расширим наш язык – будем считать, что у нас есть обозначения не только для всех рациональных чисел, но всех рациональных функций от параметров $a_1, \ldots, a_m$.

Давайте попробуем применить к формуле (9) алгоритм Тарского.

Мы можем выполнять арифметические действия с рациональными функциями, но уже первый шаг алгоритма вызывает трудности: некоторый многочлен $P(a_1, \ldots, a_m, x)$ при каких-то значениях параметров может оказаться тождественно равным нулю и не быть таким при другом выборе значений параметров. Чтобы иметь возможность выполнять алгоритм, мы будем строить *дерево разбора случаев*. Каждый раз, когда для какого-то выражения (например, коэффициента при наибольшей степени x в каком-либо многочлене) надо знать его знак или то, что он отличен от нуля, мы будем рассматривать три подслучая: это выражение меньше нуля, равно нулю, больше нуля. В результате возникнет большое дерево случаев, подслучаев, подподслучаев, ..., в каждой ветви которого мы доведём выполнение алгоритма до конца и узнаем, истинна или ложна формула при сделанных предположениях относительно параметров. Чтобы получить бескванторную формулу, эквивалентную формуле (9), нам остаётся выбрать все ветви, где ответом является ИСТИНА, конъюнктивно объединить все предположения, сделанные вдоль каждой такой ветви, и затем дизъюнктивно объединить получившиеся конъюнкции.

Теперь можно описать применение алгоритма Тарского к произвольной замкнутой формуле Ψ языка $\mathcal{A}$. Если в ней есть кванторы, то выберем среди них один такой, в области действия которого нет других кванторов. С этого квантора начинается некоторая формула вида (9). Заменив в Ψ эту формулу на эквивалентную ей бескванторную формулу, мы получим формулу, эквивалентную формуле Ψ, но имеющую на один квантор меньше. Продолжая этот процесс, мы получим в конце концов формулу без переменных, установление истинности или ложности которой сводится к простому вычислению.

Описанную выше версию алгоритма Тарского запрограммировать несложно, однако даже для простых формул программа будет работать очень долго и требовать огромной памяти. Тот же недостаток имел и первоначальный вариант, предложенный самим Тарским – время работы алгоритма нельзя было ограничить никакой башней экспонент от длины формулы. С тех пор многие исследователи улучшали алгоритм, в частности, Г.Е. Коллинз предложил алгоритм, основанный на *цилиндрической алгебраической декомпозиции* (*cylindrical algebraic decomposition*), время работы которого ограничено двойной экспонентой. Прогресс достигнут за счёт того, что при элиминации кванторов строится эквивалентная бескванторная формула в языке более широком, чем наш язык $\mathcal{A}$. К сожалению, существенного дальнейшего улучшения в общем случае ожидать не приходится – было доказано, что *любой* алгоритм будет работать дважды экспоненциальное время на некоторых «плохих» формулах. Тем не менее, построение цилиндрической алгебраической декомпозиции реализовано в ряде систем компьютерной алгебры и с успехом применяется к конкретным формулам.

Литература

1. *A.Tarski.* A decision method for elementary algebra and geometry} Santa Monica CA: RAND Corp., 1948.
2. *http://en.wikipedia.org/wiki/Tarski*
3. *http://ru.wikipedia.org/wiki/Тарский,_Альфред*
4. *http://en.wikipedia.org/wiki/Quantifier_elimination*
5. *http://en.wikipedia.org/wiki/Tarski's_axioms*
6. *http://plus.maths.org/issue23/features/kissing/index.html*

Выдающийся математический логик Альфред Тарский родился в 1901 году в Варшаве.

Алгоритм для установления истинности замкнутых формул с вещественными переменными он придумал в начале 30-х годов, но первоначально этот результат был сформулирован в других терминах. В 1931 другой великий математический логик Курт Гёдель опубликовал революционную работу, в которой показал, что в формальной арифметике могут быть верные, но недоказуемые утверждения. Результат Тарского был в некотором смысле противоположным: он показал, что если все переменные принимают вещественные значения (а не целочисленные как у Гёделя), то можно построить полную систему аксиом – какова бы ни была замкнутая формула, из этих аксиом можно вывести либо саму формулу, либо её отрицание. Отсюда, конечно, получается такой алгоритм: будем выводить из аксиом всевозможные следствия и ждать, пока не выведется либо интересующая нас формула (и тогда она истинна), либо её отрицание (и тогда наша формула является ложной).

Прошло очень много времени, прежде чем результат был опубликован. В 1939 году Тарский был вынужден эмигрировать в США, но и там из-за наступившей войны не смог сразу напечатать свой замечательный алгоритм. Тарский сделал это только в 1948 году, причем в виде отдельной книги – столь сложным был первоначальный вариант алгоритма.

Ученица Тарского Джулия Робинсон установила, что в случае, когда допустимыми значениями переменных являются рациональные числа, не существует алгоритма для распознавания истинности формул. Таким образом, алгоритм Тарского представляется счастливым исключением на фоне других похожих, но алгоритмически неразрешимых проблем.

Abstract

Tarski's algorithm allows us to determine whether a given statement about a finite set of real numbers is true of false. Together with the Cartesian coordinate system, it allows us to prove automatically a large class of theorems of elementary geometry.

The version of the algorithm presented here is suitable for introduction to the subject it is not difficult to understand the algorithm, it is easy to write a program, but it would be extremely inefficient.

Матиясевич Юрий Владимирович,
академик РАН,
зав. лабораторией математической логики ПОМИ РАН
http://logic.pdmi.ras.ru/~yumat