Tarski Algorithm

Kirill Shmakov

Monday, March 26, 2007

Abstract

This paper considers the Tarski Algorithm and its implementation to the finite geometrical problems.

Contents

1 Introduction

1.1 Motivation

"...the most ignorant Person at a reasonable Charge, and with a little bodily Labour, may write Books in Philosophy, Poetry, Politicks, Law, Mathematics and Theology, without the least Assistance from Genius or Study."
Jonathan Swift — Gulliver's Travels

Problem of generation of correct assertion is well-known side of computer usage in proof theory. Another side is verification of concrete assertion. We are going to consider one special type of assertions - the systems of polynomial inequalities and equations. Then we apply such systems to geometry for solving finite geometrical problems. Main idea belongs to the polish mathematician Alfred Tarski.

1.2 A. Tarski

Figure 1: A. Tarski

Alfred Tarski (January 14, 1902, Warsaw, Russian-ruled Poland — October 26, 1983, Berkeley, California) was a logician and mathematician who spent four decades as a professor of mathematics at the University of California, Berkeley. He was born Alfred Teitelbaum, to parents who were Polish Jews in Warsaw. Tarski's first paper, published when he was only 19 years old, was on set theory. He was a youngest person ever to complete a doctorate at Warsaw University. Tarski left Poland in August 1939, on the last ship to sail from Poland for the United States before the German invasion of Poland and the outbreak of World War II. He became an American citizen in 1945. Tarski wrote papers on topology, geometry, measure theory, mathematical logic, set theory, metamathematics, and above all, model theory, abstract algebra, and algebraic logic. During his life he supervised 24 Ph.D. dissertations, 5 by women.

2 Formalization

2.1 Geometrical view

We can mark out one kind of geometrical problems among other: problems on proof. In such problems we used to prove given assertion. Here is an example of assertion: that points of intersection of medians, heights and perpendicular bisectors lies on the same line.

We should to find the way of formolization of geometrical problems due to aplication to authomatic proofs. We describe the language of geometry. First we simplify the picture by introducing some geometrical objects:

- points
- lines
- circles

and so on. This amount of concepts is insufficient for formulating of assertions. Among objects we should also use relations between them. Let us introduce some relations and their formal denotions in terms of predicats:

- "Point A is on the line l", $\mathrm{OnLine}(A,l)$
- "Point A is on the circle O", $\mathrm{OnCircle}(A,O)$
- "The distance between A and B equals distance between C and D", $\mathrm{EqDistance}(A,B,C,D)$

Now we can formulate some axioms in this terms:

- "For any points A, B there are exists line l, such as A and B are on l"

$$\leftrightarrow \forall A \forall B \exists l \{\mathrm{OnLine}(A,l) \& \mathrm{OnLine}(B,l)\}$$

- "If points A and B both lies on lines l and m, and if A and B are different, then l and m coincides."

$$\leftrightarrow \forall A \forall B \forall l \forall m \{A \neq B \& \mathrm{OnLine}(A,l) \& \mathrm{OnLine}(B,l) \& \\ \& \mathrm{OnLine}(A,m) \& \mathrm{OnLine}(B,m) \Rightarrow l = m\}$$

And of course our main goal is formalization of assertions of problems on proof. For example, the well known

Proposition. *Medians of triangle intersects at one point.*

now can be reformulated with introduced notions as:

Proposition. *For any three mutually different points A_1, A_2 and A_3 there are four points B_1, B_2, B_3 and C and six lines l_1, l_2, l_3, m_1, m_2 and m_3 such*

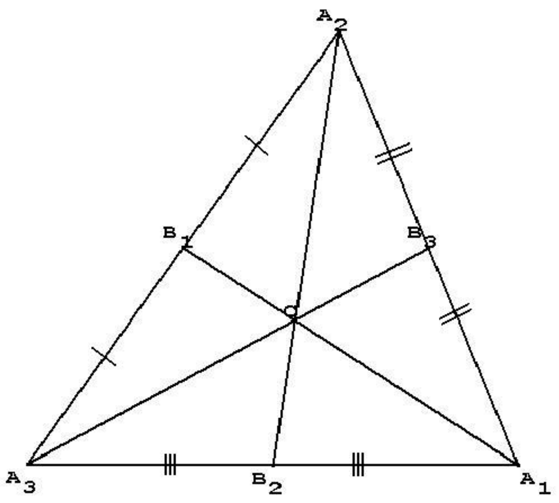

Figure 2: Medians intersects at one point

as:

$$
\begin{gathered}
\mathrm{OnLine}(A_2, l_1)\&\mathrm{OnLine}(A_3, l_1)\&\mathrm{OnLine}(B_1, l_1)\&\\
\mathrm{OnLine}(A_1, l_2)\&\mathrm{OnLine}(A_1, l_2)\&\mathrm{OnLine}(B_2, l_2)\&\\
\mathrm{OnLine}(A_1, l_3)\&\mathrm{OnLine}(A_2, l_3)\&\mathrm{OnLine}(B_3, l_3)\&\\
\mathrm{OnLine}(A_1, m_1)\&\mathrm{OnLine}(B_1, m_1)\&\mathrm{OnLine}(C, m_1)\&\\
\mathrm{OnLine}(A_2, m_2)\&\mathrm{OnLine}(B_2, m_2)\&\mathrm{OnLine}(C, m_2)\&\\
\mathrm{OnLine}(A_3, m_3)\&\mathrm{OnLine}(B_3, m_3)\&\mathrm{OnLine}(C, m_3)\&\\
\mathrm{EqDistance}(A_1, B_2, B_2, A_3)\&\\
\mathrm{EqDistance}(A_2, B_1, B_1, A_3)\&\\
\mathrm{EqDistance}(A_1, B_3, B_3, A_2)
\end{gathered}
$$

In order to make theory mory easy we shorten the list of objects. Namely we leave only points. Neveretheless we can still formulate relations using predicats which works only with points. Instead of considering complicated objects we consider their points.

- "Points A, B and C are on the same line" OnLine(A, B, C)
- "Points A and B are on the same circle with center C" OnCircle(A, B, C)
- "The distance between A and B equals distance between C and D" EqDistance(A, B, C, D)

And our example converts to

Proposition. *For any three points A_1, A_2 and A_3 there are four points B_1,*

B_2, B_3 and C such as:

$$\begin{aligned}&A_1 \neq A_2 \& A_1 \neq A_3 \& A_2 \neq A_3 \Rightarrow\\ &\quad \mathrm{OnLine}(A_1, A_2, B_3)\&\mathrm{OnLine}(A_2, A_3, B_1)\&\mathrm{OnLine}(A_1, A_3, B_2)\&\\ &\quad \mathrm{OnLine}(A_1, B_1, C)\&\mathrm{OnLine}(A_2, B_2, C)\&\mathrm{OnLine}(A_3, B_3, C)\&\\ &\quad \mathrm{EqDistance}(A_2, B_1, B_1, A_3)\&\\ &\quad \mathrm{EqDistance}(A_1, B_2, B_2, A_3)\&\\ &\quad \mathrm{EqDistance}(A_1, B_3, B_3, A_2)\end{aligned}$$

2.2 Algebraical view

Since we make assertions acceptable for machines we should exclude such incomprehensible things such as points and predicats working on points from our model. Broadly speaking machines understands only rational numbers, so we should learn how to write propositions in it.

First we replace points by their coordinates: point $A \leftrightarrow n$tiple $(x_1, ..., x_n)$ of reals

In case of plane it is convenient to use (x, y) instead of (x_1, x_2). We recur to our example:

Proposition. *For any real numbers $a_{1,x}$, $a_{1,y}$, $a_{2,x}$, $a_{2,y}$, $a_{3,x}$, $a_{3,y}$, there are such real $b_{1,x}$, $b_{1,y}$, $b_{2,x}$, $b_{2,y}$, $b_{3,x}$, $b_{3,y}$, c_x and c_y:*

$$\begin{aligned}&(a_{1,x} \neq a_{2,x} \vee a_{1,y} \neq a_{2,y})\&(a_{1,x} \neq a_{3,x} \vee a_{1,y} \neq a_{3,y})\&\\ &\quad (a_{2,x} \neq a_{3,x} \vee a_{2,y} \neq a_{3,y}) \Rightarrow\\ &\quad \mathrm{OnLine}(a_{1,x}, a_{1,y}, a_{2,x}, a_{2,y}, b_{3,x}, b_{3,y})\&\\ &\quad \mathrm{OnLine}(a_{2,x}, a_{2,y}, a_{3,x}, a_{3,y}, b_{1,x}, b_{1,y})\&\\ &\quad \mathrm{OnLine}(a_{1,x}, a_{1,y}, a_{3,x}, a_{3,y}, b_{2,x}, b_{2,y})\&\\ &\quad \mathrm{OnLine}(a_{1,x}, a_{1,y}, b_{1,x}, b_{1,y}, c_x, c_y)\&\\ &\quad \mathrm{OnLine}(a_{2,x}, a_{2,y}, b_{2,x}, b_{2,y}, c_x, c_y)\&\\ &\quad \mathrm{OnLine}(a_{3,x}, a_{3,y}, b_{3,x}, b_{3,y}, c_x, c_y)\&\\ &\quad \mathrm{EqDistance}(a_{1,x}, a_{1,y}, b_{2,x}, b_{2,y}, b_{2,x}, b_{2,y}, a_{3,x}, a_{3,y})\&\\ &\quad \mathrm{EqDistance}(a_{2,x}, a_{2,y}, b_{1,x}, b_{1,y}, b_{1,x}, b_{1,y}, a_{3,x}, a_{3,y})\&\\ &\quad \mathrm{EqDistance}(a_{1,x}, a_{1,y}, b_{3,x}, b_{3,y}, b_{3,x}, b_{3,y}, a_{2,x}, a_{2,y})\end{aligned}$$

Predicats can be simplified to numeric terms now:

$$\begin{aligned}&\mathrm{OnLine}(a_x, a_y, b_x, b_y, c_x, c_y) \longleftrightarrow\\ &\qquad a_x b_y + a_y c_x + b_x c_y - a_x c_y - a_y b_x - b_y c_x = 0\end{aligned}$$

$$\begin{aligned}&\mathrm{EqDistance}(a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y) \longleftrightarrow\\ &\qquad (a_x - b_x)^2 + (a_y - b_y)^2 = (c_x - d_x)^2 + (c_y - d_y)^2\end{aligned}$$

And now, only in terms of trivial operations, we receive:

Proposition. *For any real numbers $a_{1,x}$, $a_{1,y}$, $a_{2,x}$, $a_{2,y}$, $a_{3,x}$, $a_{3,y}$, there are such real $b_{1,x}$, $b_{1,y}$, $b_{2,x}$, $b_{2,y}$, $b_{3,x}$, $b_{3,y}$, c_x and c_y:*

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

It's not so easy to recognize in this proposition the well-known fact from elementary school.

3 Algorithm

3.1 Tarski Theorem

Let us introduce special language A, which contains

- *notation* for all rational numbers
- *variables* for real numbers
- *operations* of addition and multiplication for constructing polynomials
- *unary predicates* $= 0$, > 0, < 0, so the elementary formulas have forms $\mathrm{P} = 0$, $\mathrm{P} > 0$, and $\mathrm{P} < 0$
- *logical connectives* $\&$, $\vee$, $\neg$, $\Rightarrow$
- *quantifiers* $\forall$ and $\exists$

This ingredients of language A are necessary for constructing the systems of polynomial inequalities and equations, witch we are going to verify. Now we specify the meaning of "system of polynomial inequalities and equations".

We can't called formula

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

assertion, because its verity depends on values of variables. Although we can ask different precize questions:

- Is it true for $x = 4$ and $y = 5$?
- Is it true for any x and y?
- Do such x and y exist?

Recall some logical notions.

Definition. 1. *Open formulla is correctly constructed formula without quantifiers.*

2. *Partially open formula is correctly constructed formula wich contain open and closed variables.*

3. *Closed formula is correctly constructed formula where each entring variable covered by corresponding quantifier.*

And corresponding

Example. 1. $(x \equiv y) \to (z \& (y \vee \neg x))$

2. $\forall x (\exists y (x \vee y) \& (\forall z ((x \& y) \to z)))$

3. $\forall x \exists y (\forall z (x \vee y \vee z) \& \exists z (x \& \neg y \& \neg z))$

We are going to verfy the systems of closed polynomial inequalities and equations with the following

Theorem. *(Alfred Tarski) There exists an algorithm for deciding for a given arbitrary closed formula of the language A whether the formula is true or not.*

This algorithm may verify for example following formula

$$\forall a_{1,x} \forall a_{1,y} \forall a_{2,x} \forall a_{2,y} \forall a_{3,x} \forall a_{3,y} \exists b_{1,x} \exists b_{1,y} \exists b_{2,x} \exists b_{2,y} \exists b_{3,x} \exists b_{3,y} \exists c_x \exists c_y :$$

$$\begin{gathered}
((a_{1,x} - a_{2,x})^2 > 0 \vee ((a_{1,y} - a_{2,y})^2 > 0) \& ((a_{1,x} - a_{3,x})^2 > 0 \vee ((a_{1,y} - a_{3,y})^2 > 0) \& \\
((a_{2,x} - a_{3,x})^2 > 0 \vee ((a_{2,y} - a_{3,y})^2 > 0) \Rightarrow \\
a_{1,x}a_{2,y} + a_{1,y}b_{3,x} + a_{2,x}b_{3,y} - a_{1,x}b_{3,y} - a_{1,y}a_{2,x} - a_{2,y}b_{3,x} = 0 \ \& \\
a_{2,x}a_{3,y} + a_{2,y}b_{1,x} + a_{3,x}b_{1,y} - a_{2,x}b_{1,y} - a_{2,y}a_{3,x} - a_{3,y}b_{1,x} = 0 \ \& \\
a_{1,x}a_{3,y} + a_{1,y}b_{2,x} + a_{3,x}b_{2,y} - a_{1,x}b_{2,y} - a_{1,y}a_{3,x} - a_{3,y}b_{2,x} = 0 \ \& \\
a_{1,x}b_{1,y} + a_{1,y}c_x + b_{1,x}c_y - a_{1,x}c_y - a_{1,y}b_{1,x} - b_{1,y}c_x = 0 \ \& \\
a_{2,x}b_{2,y} + a_{2,y}c_x + b_{2,x}c_y - a_{2,x}c_y - a_{2,y}b_{2,x} - b_{2,y}c_x = 0 \ \& \\
a_{3,x}b_{3,y} + a_{3,y}c_x + b_{3,x}c_y - a_{3,x}c_y - a_{3,y}b_{3,x} - b_{3,y}c_x = 0 \ \& \\
(a_{1,x} - b_{2,x})^2 + (a_{1,y} - b_{2,y})^2 = (b_{2,x} - a_{3,x})^2 + (b_{2,y} - a_{3,y})^2 \ \& \\
(a_{2,x} - b_{1,x})^2 + (a_{2,y} - b_{1,y})^2 = (b_{1,x} - a_{3,x})^2 + (b_{1,y} - a_{3,y})^2 \ \& \\
(a_{1,x} - b_{3,x})^2 + (a_{1,y} - b_{3,y})^2 = (b_{3,x} - a_{2,x})^2 + (b_{3,y} - a_{2,y})^2
\end{gathered}$$

3.2 Proof

We will proof proposition of theorem by induction on number of variables in system. First we precisely consider the case of one-variable system and then show less formaly the idea of induction step.

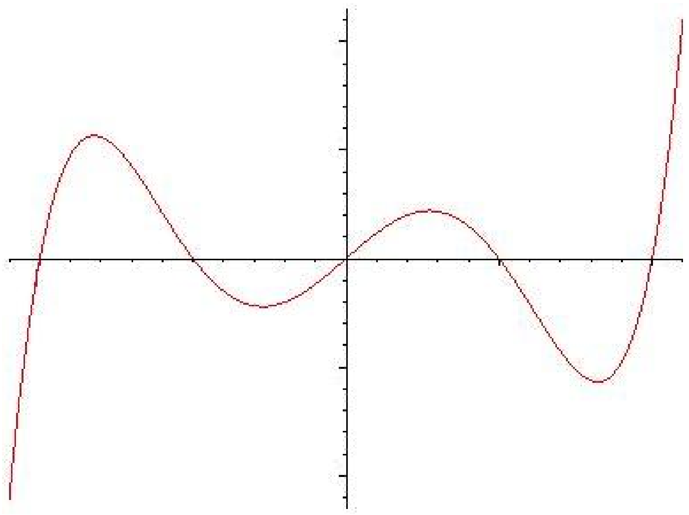

Figure 3: Graph of random polynomial

3.2.1 Base

Let all polynomials are one variable. We display graph of random polynomial f for obviousness(fig.3).

For example, if we interested in areas where $f(x) > 0$:

As you can see(fig.4) the verity of inequality changes only when we come throw the roots of polynomial. So, between the roots the verity of system stay the constant, and if we want to learn all posibilities it's enough to check system only in points which lie on the segments between adjacent roots and in two additional points: one which lefter than all roots and another one which righter.

After spliting the system into separate polynomials, we can check them one by one. Now we can sketch a skeleton of algorithm for one-variable case:

"Algorithm" of Tarski {version 0.1} for formula $Qx\Phi(x)$

1. Produce the list $\mathrm{P}_1(x), \dots, \mathrm{P}_\mathrm{k}(x)$ of all polynomials which occur in $\Phi(x)$
2. Compute the set $\mathfrak{N} = \{x_0, \dots, x_n\}$ consiting of all real roots of all polynomials $\mathrm{P}_1(x), \dots, \mathrm{P}_\mathrm{k}(x)$ which are different from identical zero; assume $x_0 < x_1 < \dots < x_{n-1} < x_n$
3. Extend the set $\mathfrak{N}$ to the set $\mathfrak{M} = \{y_0, \dots, y_m\} \supset \mathfrak{N}$ such that
   - for every i, such that $0 < i \leqslant n$, there exists j, such that $0 < i \leqslant n$ and $x_{i-1} < y_j < x_i$
   - for every i, such that $0 \leqslant i \leqslant m$, $y_0 < x_i$
   - for every i, such that $0 \leqslant i \leqslant n$, $x_i < y_m$
4. Formula $\exists x\Phi(x)$ is true if and only if $\Phi(y_0) \bigvee \dots \bigvee \Phi(y_m)$.
   Formula $\forall x\Phi(x)$ is true if and only if $\Phi(y_0) \& \dots \& \Phi(y_m)$.

We name it "algorithm" because it contains number of steps which we unable to programme. We try gradually throw them out of construction. First we simplify third step. Instead of accomplishment of new action such as

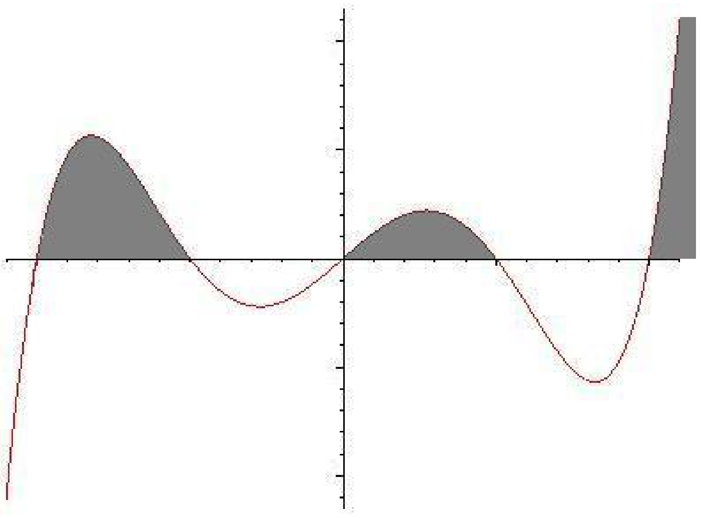

Figure 4: Areas of verity $f(x) > 0$

extension of set of roots we just repeat old one. We calculate more roots and use redundant roots exactly as points y_i in described model.

Recall simple fact from calculus, that between every two roots of function there is a root of derivative(fig.6). After that we easily understand, that it's sufficient to check roots of derivative as the points y_i. As we have number of polynomials we should take into consideration all intervals between adjacent roots of all polynomials. The most easy construction which considers roots of all polynomials is product of this polynomials. So, it's enough to add roots of derivative of product of all polynomials to deal with inner intervals. For investigation two extreme rays it's enough to add two points, broadly speaking, $-\infty$ and $+\infty$.

From here we come to the next version of "algorithm":

"Algorithm" of Tarski {version 0.2} for formula $Qx\Phi(x)$

1. Produce the list $\mathrm{P_1}(x), \ldots, \mathrm{P_k}(x)$ of all polynomials which occur in $\Phi(x)$ ($\mathrm{P_i}(x) \not\equiv 0$)
2. Add the polynomial $\mathrm{P_0}(x) = (\mathrm{P_1}(x) \ldots \mathrm{P_k}(x))'$
3. Compute the set $\mathfrak{N} = \{x_0, \ldots, x_n\}$ consiting of all real roots of all polynomials $\mathrm{P_0}(x), \mathrm{P_1}(x), \ldots, \mathrm{P_k}(x)$ which are different from identical zero
4. Extend the set $\mathfrak{N}$ to the set $\mathfrak{M} = \{x_{-\infty}, x_0, x_1, \ldots, x_n, x_{+\infty}\}$ where $x_{-\infty}$ and $x_{+\infty}$ are such numbers that $x_{-\infty} < x_0 < x_1 < \cdots < x_{n-1} < x_n < x_{+\infty}$
5. Formula $\exists x\Phi(x)$ is true if and only if

$$\Phi(x_{-\infty}) \vee \Phi(x_0) \vee \cdots \vee \Phi(x_n) \vee \Phi(x_{+\infty})$$

Formula $\forall x\Phi(x)$ is true if and only if

$$\Phi(x_{-\infty})\&\Phi(x_0)\&\ldots\&\Phi(x_n)\&\Phi(x_{+\infty})$$

For the following modification we introduce a notion of Tarski table.

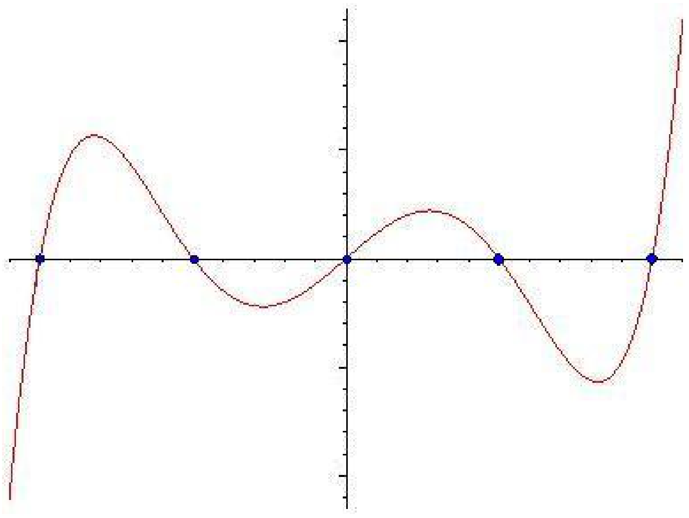

Figure 5: Significant points

Definition. *Tarski table is table of values of all polynomials in all points.*

	$x_{-\infty}$	x_0	$\dots$	x_j	$\dots$	x_n	$x_{+\infty}$
$\mathrm{P_0}(x)$	$\mathrm{P_0}(x_{-\infty})$	$\mathrm{P_0}(x_0)$	$\dots$	$\mathrm{P_0}(x_j)$	$\dots$	$P_0(x_n)$	$P_0(x_{+\infty})$
$\vdots$							
$\mathrm{P_i}(x)$	$\mathrm{P_i}(x_{-\infty})$	$\mathrm{P_i}(x_0)$	$\dots$	$\mathrm{P_i}(x_j)$	$\dots$	$\mathrm{P_i}(x_n)$	$\mathrm{P_i}(x_{+\infty})$
$\vdots$							
$\mathrm{P_k}(x)$	$\mathrm{P_k}(x_{-\infty})$	$\mathrm{P_k}(x_0)$	$\dots$	$\mathrm{P_k}(x_j)$	$\dots$	$\mathrm{P_k}(x_n)$	$\mathrm{P_k}(x_{+\infty})$

We assume that $x_{-\infty} < x_0 < \cdots < x_j \cdots < x_n < x_{+\infty}$

Tarski table have some significant for us properties. As we remember, all x_j was added in set as roots of some polynomial, so every point is someone's root:

$$\forall j \exists i \{\mathrm{P_i}(x) \not\equiv 0 \& \mathrm{P_i}(x_j) = 0\}$$

and all roots are added:

$$\forall i \forall x' \{(\mathrm{P_i}(\mathrm{x}) \not\equiv 0 \& \mathrm{P_i}(\mathrm{x}') = 0) \Rightarrow \exists j \{x' = x_j\}\}$$

If we imagine that we can construct Tarski table than we can rewrite "algorithm" in the next way:

"Algorithm" of Tarski {version 0.3} for formula $Qx\Phi(x)$

1. Produce the list $\mathrm{P_1(x)}, \dots, \mathrm{P_k(x)}$ of all polynomials which occur in $\Phi(x)$ ($\mathrm{P_i}(x) \not\equiv 0$)
2. Add the polynomial $\mathrm{P_0}(x) = (\mathrm{P_1}(x) \dots \mathrm{P_k}(x))'$
3. Construct Tarski table for $\mathrm{P_0}(x), \mathrm{P_1}(x), \dots, \mathrm{P_k}(x)$
4. Calculate logical values $\Phi(x_{-\infty})$, $\Phi(x_0)$, $\dots$, $\Phi(x_n)$, $\Phi(x_{+\infty})$ using the values of the polynomials from the table (but not the values of the x's)

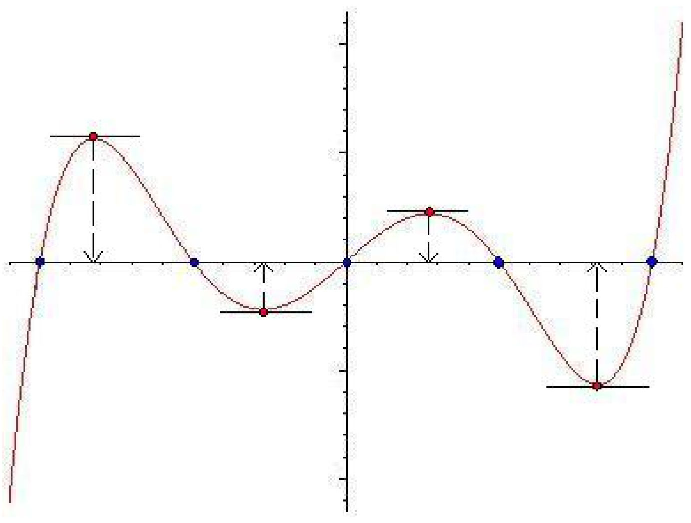

Figure 6: Roots structure

5. Formula $\exists x \Phi(x)$ is true if and only if

$$\Phi(x_{-\infty}) \vee \Phi(x_0) \vee \cdots \vee \Phi(x_n) \vee \Phi(x_{+\infty})$$

Formula $\forall x \Phi(x)$ is true if and only if

$$\Phi(x_{-\infty}) \& \Phi(x_0) \& \ldots \& \Phi(x_n) \& \Phi(x_{+\infty})$$

We are still unable to find roots of polynomials, but for now all other steps can be programmed pretty good. First we notice that all possible predicats come from comparison of polynomials with null. So we are not interested in numerical values of polynomials in point, we are interested only in their signs.

From here we come to

Definition. *Semisimplified Tarski table for polynomials* $\mathrm{P_0}(x), \ldots, \mathrm{P_k}(x)$ *is table which obtains from simple Tarski table with cancellation of numerical values.*

	$-\infty$	x_0	$\ldots$	x_j	$\ldots$	x_n	$+\infty$
$\mathrm{P_0}(x)$							
$\vdots$							
$\mathrm{P_i}(x)$				t_{ij}			
$\vdots$							
$\mathrm{P_k}(x)$							

$$t_{ij} = \begin{cases} -, & \textit{if } \mathrm{P_i}(x_j) < 0 \\ 0, & \textit{if } \mathrm{P_i}(x_j) = 0 \\ +, & \textit{if } \mathrm{P_i}(x_j) > 0 \end{cases}$$

In the algorithm we are not interested in points except the step, where we calculate the Tarski table, bur it's turn out that for constructing of body of semisimplified Tarski table not necessary know the values of x_i.

Definition. *Simplified Tarski table for polynomials* $\mathrm{P}_0(x), \ldots, \mathrm{P}_\mathrm{k}(x)$ *is just semisimplified table without values of* x_i*.*

$\mathrm{P}_0(x)$	$\pm\, 0$	$\pm\, 0$	$\ldots$	$\pm\, 0$	$\ldots$	$\pm\, 0$	$\pm\, 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathrm{P}_\mathrm{i}(x)$	$\pm\, 0$	$\pm\, 0$	$\ldots$	$\pm\, 0$	$\ldots$	$\pm\, 0$	$\pm\, 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathrm{P}_\mathrm{k}(x)$	$\pm\, 0$	$\pm\, 0$	$\ldots$	$\pm\, 0$	$\ldots$	$\pm\, 0$	$\pm\, 0$

For this we need enlarge our set of polynomials. And after construction of simplified Tarski table for them, we automatically receive simplified Tarski table for initial set of polynomials.

Definition. *A system of functions is called semisaturated, if with each function the system contains its derivative.*

It's obvious exercise to prove the next

Lemma. *Every finite system of polynomials can be extended to a finite semisaturated system of polynomials.*

For the semisaturate system of polynomials faithful the following

Lemma. *If the system of polynomials* $\mathrm{P}_0(x), \ldots, \mathrm{P}_\mathrm{k}(x)$ *is semisaturated and* $\mathrm{P}_\mathrm{i}(x) \not\equiv 0$*, then the ith row cannot contain* 0 *in two consequetive cells.*

Proposition of lemma easy follows from our remark about root of derivative between roots of function. For the final cut we introduce the

Definition. *A semisaturated system of polynomials* $\mathrm{P}_0(x), \ldots, \mathrm{P}_\mathrm{n}(x)$ *is called saturated if for each its two polynomials* $\mathrm{P}_\mathrm{k}(x)$ *and* $\mathrm{P}_\mathrm{l}(x)$ *such that*

$$0 < \deg(\mathrm{P}_\mathrm{l}(\mathrm{x})) \leqslant \deg(\mathrm{P}_\mathrm{k}(x)),$$

the system also contains the remainder $\mathrm{R}(x)$ *from dividing* $\mathrm{P}_\mathrm{k}(x)$ *by* $P_l(x)$*, i.e.,*

$$\mathrm{P}_\mathrm{k}(x) = \mathrm{Q}(x)\mathrm{P}_\mathrm{l}(x) + \mathrm{R}(x), \deg(\mathrm{R}(x)) < \deg(\mathrm{P}_\mathrm{l}(x))$$

The following lemma gives us possibility to extend our initial system of polynomials to saturated system.

Lemma. *Every finite system of polynomials can be extended to a finite saturated system of polynomials.*

Lemma can be prooven with consequtive examination of degrees starting from the biggest.

Saturated system of polynomials poses the next property, which allow us construct it inductivly:

Lemma. *If* $\mathrm{P}_0(x), \ldots, \mathrm{P}_{\mathrm{k}-1}(x), \mathrm{P}_\mathrm{k}(x)$ *is a saturated system of polynomials and*

$$\deg(\mathrm{P}_0(x)) \leqslant \cdots \leqslant \deg(\mathrm{P}_{\mathrm{k}-1}(x)) \leqslant \deg(\mathrm{P}_\mathrm{k}(x)),$$

then the system $\mathrm{P}_0(x) \ldots \mathrm{P}_{\mathrm{k}-1}(x)$ *is also saturated.*

From this we come to inductive construction of simplified Tarski system for saturated system $\mathrm{P_0}(x), \dots, \mathrm{P_k}(x)$.

First we choose all constant polynomials from the set and construct table for them.

$\mathrm{P_0}(x)$	0	0
$\mathrm{P_1}(x)$	$\pm$	$\pm$
$\vdots$	$\vdots$	$\vdots$
$\mathrm{P_k}(x)$	$\pm$	$\pm$

Let us construct table for first $k-1$ polynomials, and we want to construct table for next one.

From representation of polynomial $\mathrm{P_k}(x) = p_n x^n + p_{n-1}x^{n-1} + \dots + p_0$ we can guess the contents of extreme cells as the sign on $-\infty$ and $+\infty$.

$\mathrm{P_0}(x)$	0	0	...	0	...	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathrm{P_i}(x)$	$\pm$	$\pm$ 0	...	$\pm$ 0	...	$\pm$ 0	$\pm$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathrm{P_{k-1}}(x)$	$\pm$	$\pm$ 0	...	$\pm$ 0	...	$\pm$ 0	$\pm$
$\mathrm{P_k}(x)$	$\pm$						$\pm$

For fulfilling of another cells we will use the saturation of system. Namely for every l exists such m so

$$\mathrm{P_k}(x) = \mathrm{Q}(x)\mathrm{P_l}(x) + \mathrm{P_m}(x)$$

If in point x_i $\mathrm{P_l}$ turns to be 0 then it's enough copy to i-th cell for $\mathrm{P_k}$ the value of i-th cell for $\mathrm{P_m}$:

$$\mathrm{P_k}(x_i) = \mathrm{Q}(x_i)\mathrm{P_l}(x_i) + \mathrm{P_m}(x_i) = \mathrm{P_m}(x_i)$$

Due to construction of the system for each x_i exists such $\mathrm{P_l} \not\equiv 0$:
$\mathrm{P_l}(x_i) = 0$, so we can fulfill all cells.

It remains to add roots of new polynomial. Because every two roots of it separated by roots of derivative which was added before, it's enough to examinate every pair of adjacent columns on apperance of root of $\mathrm{P_k}$ between them. Evidently it's appear if and only if in the corresponding adjacent cells stays plus and minus.

$\mathrm{P_0}(x)$	$\pm$ 0	$\pm$ 0	...	$\pm$0		$\pm$0	...	$\pm$ 0	$\pm$ 0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathrm{P_i}(x)$	$\pm$ 0	$\pm$ 0	...	$\pm$0		$\pm$0	...	$\pm$ 0	$\pm$ 0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathrm{P_{k-1}}(x)$	$\pm$0	$\pm$ 0	...	$\pm$0		$\pm$0	...	$\pm$ 0	$\pm$ 0
$\mathrm{P_k}(x)$	$\pm$	$\pm$0	...	$-$	0	$+$	...	$\pm$0	$\pm$

New cells in previous rows we fulfills according to majorizing value of adjacent cells: they can't be different signs due to constrution of table. If they both equals null, it's possible only in first row, else we assume the value of midle cell to the nonzero value of adjacent cells.

With this we finish recursive construction of simplified Tarski table for the saturated system of polynomials. Table for initial system is subtable of resulting table for saturated system. We know it rows, but not columns. But it doesn't matter because we can check more points: redundant points doesn't affect to the result and we come to the final version of algorithm for one variable case:

Algorithm of Tarski {version 0.4} for formula $Qx\Phi(x)$

1. Produce list $\mathrm{Q_1}(x), \dots, \mathrm{Q_l}(x)$ of all polynomials which occur in $\Phi(x)$ $(\mathrm{Q_i}(x) \not\equiv 0)$
2. Append the polynomial $\mathrm{Q_0}(x) = (\mathrm{Q_1}(x) \dots \mathrm{Q_l}(x))'$
3. Extend the list to saturated system of polynomials $\mathrm{P_0}(x), \dots, \mathrm{P_k}(x)$ and order them so that
$$\deg(\mathrm{P_0}(x)) \leqslant \dots \leqslant \deg(\mathrm{P_{k-1}}(x)) \leqslant \deg(\mathrm{P_k}(x))$$
4. Construct simplified Tarski table for $\mathrm{P_0}(x), \mathrm{P_1}(x), \dots, \mathrm{P_m}(\mathrm{x})$ for $m = 0, 1, 2, \dots, k$
5. Calculate logical value of $\Phi(x)$ for every column in the table
6. Formula $\exists x\Phi(x)$ is true if and only if at least one of the calculated values of $\Phi(x)$ was true;
Formula $\forall x\Phi(x)$ is true if and only if all calculated values of of $\Phi(x)$ were true

3.2.2 Induction step

From general formula we step by step exclude the variables until there is one variable. For one variable system we use the previous algorithm. Main idea is like we solve the equation or inequality with parameter $\mathrm{P}(a, x) > 0$ we come to the answer depending on a. But we are interested only in special properties of answer set depending on quantifier on x.

For example if we have inequality of type

$$\mathrm{A} := (\exists x : P(a, x) > 0)$$

and answer is

$$\begin{cases} \mathrm{Q_1}(a) := (a < -2),\ x \in (2; 3) \cup \{-a\} \Rightarrow \mathrm{A} = \mathrm{True} \\ \mathrm{Q_2}(a) := (-2 \le a \le 3),\ x \in [4; 7 + a] \Rightarrow \mathrm{A} = \mathrm{True} \\ \mathrm{Q_3}(a) := (3 < a),\ x \in \emptyset \Rightarrow \mathrm{A} = \mathrm{False} \end{cases}$$

Then we may assume that $\mathrm{A} \equiv \mathrm{Q_1}(a) \vee \mathrm{Q_2}(a)$. For example $\exists x : bx + c = 0$ is equivalent to $(b \neq 0 \ \vee \ (b = 0 \ \& \ c = 0))$.

And $\exists x : ax^2 + bx + c = 0$ is equivalent to

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

In general view we do the following consequetive quantifiers elemenations:

$$\left\{ \begin{array}{c} \mathrm{Q_1}x_1, \ldots, \mathrm{Q_{n-1}}x_{n-1}\mathrm{Q_n}x_n : \mathrm{P_n}(x_1, \ldots, x_{n-1}, x_n) \\ \mathrm{Q_n}x_n : \mathrm{P_n}(x_1, \ldots, x_{n-1}, x_n) \leftrightarrow \mathrm{P_{n-1}}(x_1, \ldots, x_{n-1}) \end{array} \right\}$$

$$\Downarrow$$

$$\left\{ \begin{array}{c} \mathrm{Q_1}x_1, \ldots, \mathrm{Q_{n-2}}x_{n-2}\mathrm{Q_{n-1}}x_{n-1} : \mathrm{P_{n-1}}(x_1, \ldots, x_{n-2}, x_{n-1}) \\ \mathrm{Q_{n-1}}x_{n-1} : \mathrm{P_{n-1}}(x_1, \ldots, x_{n-2}, x_{n-1}) \leftrightarrow \mathrm{P_{n-2}}(x_1, \ldots, x_{n-2}) \end{array} \right\}$$

$$\Downarrow$$

$$\vdots$$

$$\Downarrow$$

$$\mathrm{Q_1}x_1 : \mathrm{P_1}(x_1)$$

For simplity we consume consider two-variable case. Many variable case investigated in the same way. We view on every polynomial like on a polynmial of x with coefficients from ring of rational function of a. Initialy there is only polynomials of a.

$$\mathrm{P}(a, x) = \sum_{i,j} P_{i,j} a^j x^i = \sum_i \left(\sum_j P_{i,j} a^j \right) x^j$$

We start to apply the method of construction of Tarski table described above, and every time when we want to divide some rational function over another on we take in excess all possible variants of sign of it's leading coefficients. If we consider the case when the leading coefficient equals 0 we start to take in excess all posyble variants for the second one and so on. Also, in every branch of excess we remember selected values for rational function end in the case, when we can splited some new function into combination of reviewed function for which we know it's sign we don't start new excess. When we start to fullfil the row in table we also want to know the leading coefficients. There we also take in excess all possible variants. Depending on resulting table we add od don't add the corresponding branch of excess in new system.

4 Conclusion

After a number of modification this method allows to prove that in three-dimensional space we can touch one ball with 12 balls of the same radius, but not with 13.

This algorithm is strikly unefficient. It's complexity cann't be bounded to the bottom by any towers of exponents with size of input at the top.

For the geometrical problems we can use the random points method. for example if we want prove some property are faithfull for all points then we can check it for a number of random points. If all tests are succed for this points then we may assume that the property are faithfull.

For speedup of algorithm we can use the cylindrical algebraic decomposition. Main idea of this method in extension of language in which we work, but it makes the algorithm more difficult.

For example
$$x^2 + y^2 + z^2 < 1$$
turns to
$$\begin{cases} -1 < x < 1, \\ -\sqrt{1-x^2} < y < \sqrt{1-x^2}, \\ -\sqrt{1-x^2-y^2} < z < \sqrt{1-x^2-y^2} \end{cases}$$
That gives us very short number of conditions.

References

[1] D. Lazard. An improved projection for cylindrical algebraic decomposition