# THE COMPUTER TOOL FOR VERIFICATION HYPOTHESES IN PARAMETRICAL PROBLEMS SOLVING

D. Mantserov, D. Petrichenko, S. Pozdnyakov


## Summary

The article considers various aspects of using verification environment (so-called «Verifier») to support students' activity in parametrical problems solving. The Verifier was created to support high school students in their work with functions. It compares student's answer with true one and shows her counterexamples accompanied by their graphs and comments. So with Verifier's support one may improve her own solution applying various conjectures. A student should consider various conditions and different cases; therefore the Verifier's answers have to be given in a complex logical form, which was a difficulty we got over in our work. We believe that our approach opens the way for expanding types of problems, which will be useful for studying function properties.


## The study and research questions. Relevant features of the theoretical framework

Problem solution is important part of mathematics teaching methodology. In the process of finding a solution to the problem some features of creative thinking dynamics are shown (Wertheimer, [1]):

- Collision with a problem;

- Ambiguity, incompleteness of a situation;

- Refinement of infringement areas;

- Using operations to change situation.

The solving process also includes conjectures making and partial solution construction (see, e.g. Polya [3, 4]).

The heuristics play an important role in mathematical problem solving (see, e.g. Schoenfeld [2]). It is interesting to consider possibilities for technological support of heuristic activities. For doing this we should base on the

- essence of didactic task;
- psychology of intellectual activity;
- computer tools usage traits or patterns;

The main question for our research is:

*How can we support productive activity of students in solving mathematical problems with logically complex solutions via computer tools usage?*

To find the answer we will explore an idea of providing a student with examples and counterexamples during the solution process.

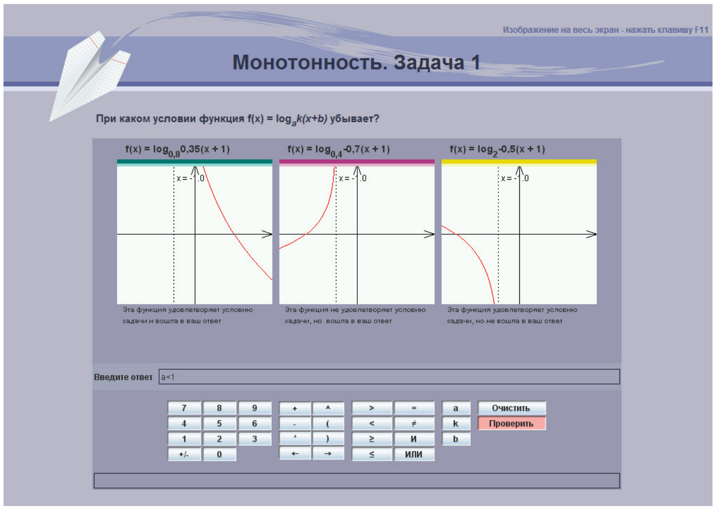
We will use M. Minsky conception [5] of two mind mechanisms for combining separate facts in new entity:

1) creating new abstraction for these facts;

2) mechanical combining of facts in new collection.

In our research we will use both mechanisms: the first one for human-computer interaction, and the second one for organizing computer support.

In problem solving process the students should construct their answers in the predicate form, so that the Verifier could check them against the set of existing examples. But one may ask why should they be of such importance for a student? First of all, it is so because abstract conceptions in human mind are closely connected with their examples ("by default") [5] and it is possible to form the new concepts only through considering “pros” and “contras” [6].




The graphical interface of Verifier is based on classical teaching routine - the "IRE sequence", which consists of three steps: a teacher initiates an interaction, a student responds then, and the teacher evaluates the response (see, e.g. Sinclair & Coulthard, 1975 [7], Cazden, 1988 [8]).

# Design and methodology

## Object classes and their properties description

Verifier’s tasks are based on parametric classes for functions such as $y=kx+b$, $y=a|x-b| + |x-c|$, $y=cx^{m/n}$, $y=\log_a k(x-b)$, $y = a^x + kb^x$ (Verifier’s environment allows to construct new classes). Every class is connected with the set of examples associated with parameters' values - which are interpreted by the Verifier to show function formulae and graphs.

Each task related to a function class may be interpreted as the class property.

Let's consider a class typical for studying in 10-11 grades of high school in Russia. It is the class of logarithmic functions which is determined by the formula $y=\log_a(bx+c)$ and a predicate which restricts the domain of parameters, for example, $a>0$ & $a\neq1$.

Various properties can be formulated as predicates of class parameters a, b, c.

*Examples.*

1) “$y=\log_a(bx+c)$ is increasing function”: ($a>1$ & $b>0$) OR ($a<1$ & $b<0$).

2) “A domain of $y=\log_a(bx+c)$ is all positive numbers”: $b>0$ & $c=0$.

3) “An intersection of function graph with y-axe is lying in upper semiplane”:

($a>1$ & $c>1$) OR ($a<1$ & $c<1$ & $c>0$).

4) “Function graph neither has points inside the III quarter of coordinate plane nor on the quarter boundaries”: ($a<1$ & $b>0$ & $c<1$ & $c>0$) OR ($a>1$ & $b<0$ & $c>1$).

One may note that the last condition may be rewrtten in other way:

(($a<1$ & $b>0$) OR ($a>1$ & $b<0$)) & (($a>1$ & $c>1$) OR ($a<1$ & $c<1$ & $c>0$)),

It means that this property can be expressed through previous ones:

“Function graph neither has points inside the III quarter of coordinate plane nor on the quarter boundaries” can be written as

(“$y=\log_a(bx+c)$ is increasing function”) & (“Function graph neither has points inside the III quarter of coordinate plane nor on the quarter boundaries”)

The last example shows that after describing some set of properties via predicates we can verify the connections between these properties.

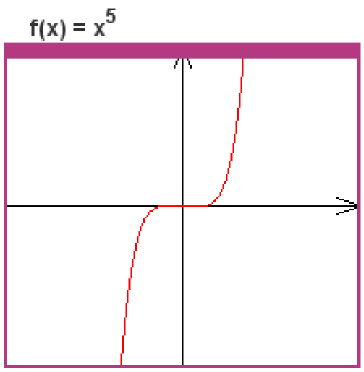
**Examples of interactive answer detailing for Verifier problems**

Let's consider two basic examples and demonstrate various logical paths while solving problems with Verifier.

*Problem. Find conditions for fraction power function* $y=cx^{m/n}$ *with the range of all positive real numbers (m and n are coprime numbers, n>0).*

*First trial*

*Presumable student reasonings.* «We know that fraction power function $y=x^{m/n}$ has only positive values, therefore c must be positive».

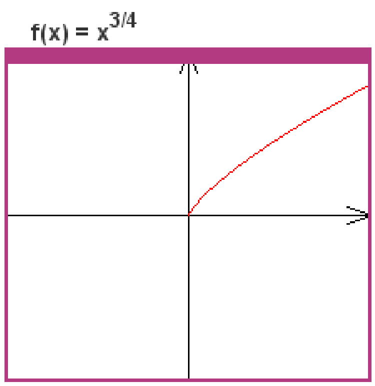



*Input:* c>0

*Output. Your answer has a mistake. Look at the picture. This function satisfies your condition but doesn't satisfy the task condition.*

*Judgment: "We need to narrow fraction power function range. So we must narrow power function's domain. To narrow fraction power function's domain we need to make it integer power function".*

*Second trial*

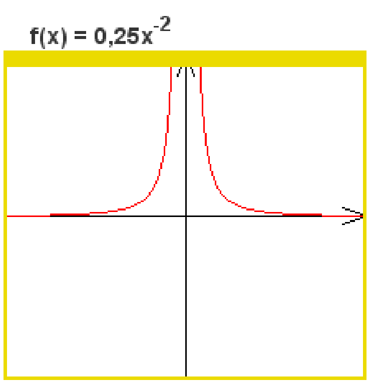



*Input:* c>0 & n>1

*Output. Your answer has a mistake. Look at the picture. This function satisfies your condition but doesn't satisfy the task condition.*

*Judgment: «Yes, of course. The function must not be defined for x=0 therefore m must be negative».*

*Third trial*

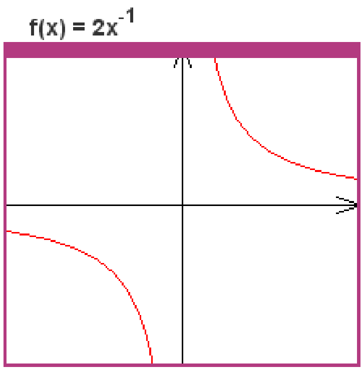



*Input:* c>0 & n>1 & m<0

*Output. Your answer has a mistake. Look at the picture. This function satisfies task condition but doesn't satisfy your condition.*

*Judgment: "Really. Narrowing fraction power function domain can be done without narrowing the function domain. The function take the same value twice. Let's return to our hypothesis of integer power functions".*

*Fourth trial*

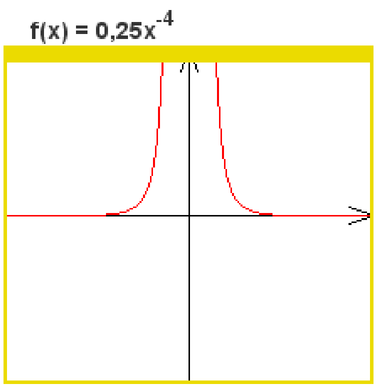



*Input:* c>0 & m<0& n≥1

*Output. Your answer has a mistake. Look at the picture. This function satisfies your condition but doesn't satisfy the task condition.*

*Judgment: "It is clear the function must be odd. Let's try to check the idea by taking m=-2".*

*Fifth trial*

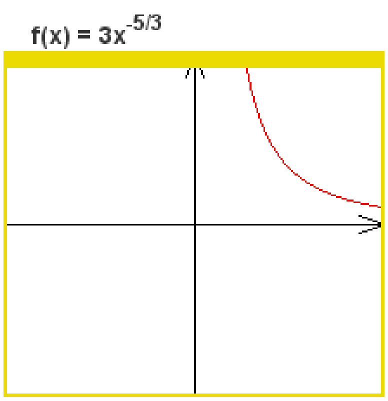



*Input:* c>0 & m<0 & n≥1 & m=-2

*Output. Your answer has a mistake. Look at the picture. This function satisfies task condition but doesn't satisfy your condition.*

*Judgment: "Yes the conjecture proves to be true. To make function even we must take even m".*

*Sixth trial*

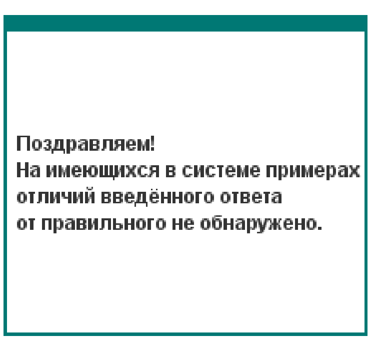



*Input:* c>0 & m<0 & n≥1& m_is_even

*Output. Your answer has a mistake. Look at the picture. This function satisfies task condition but doesn't satisfy your condition.*

*Judgment: "Ok. It appears that first idea works for functions with domain of all positive numbers, and the second one works for functions with domain of all numbers except zero. We must combine both ideas in a more accurate way".*

*Seventh trial*

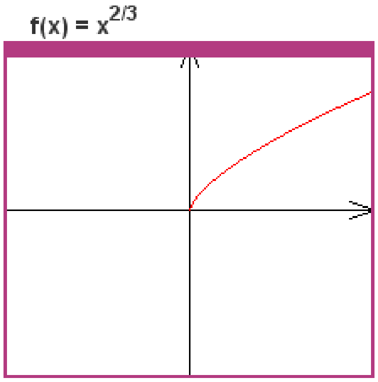



*Input:* c>0 & m<0 & (n>1 | n=1 & m_is_even)

*Output. Congratulations! It is not possible to find difference between your answer and true one using accessible examples.*

Let's consider another way of deduction while solving the same problem (it was proposed by a student).

*Problem. Find conditions for fraction power function $y=cx^{m/n}$ to have range of all positive real numbers (m and n are coprime numbers, n>0).*



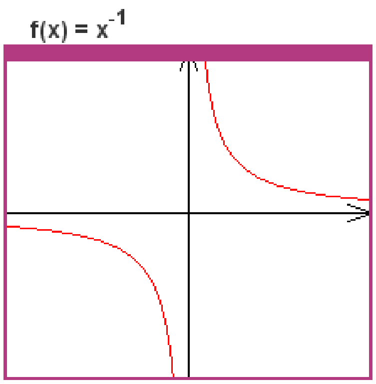

*First trial*

*Presumable student reasonings.* «We know that fraction power function $y=x^{m/n}$ has only positive values therefore c must be positive».

*Input:* c>0

*Output. Your answer has a mistake. Look at the picture. This function satisfies your condition but doesn't satisfy the task condition (here we see another counterexample).*

*Judgment: "Why this example doesn't satisfy the task condition? The range does contain zero! How to eliminate it? We need to do power parameter negative!".*

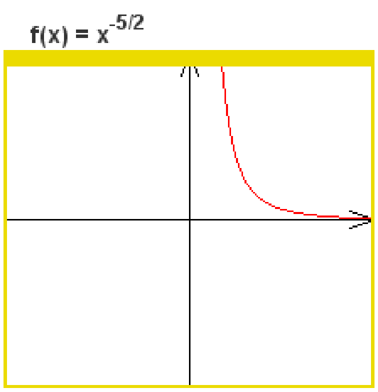



*Second trial*

*Input:* c>0 & m<0

*Output. Your answer has a mistake. Look at the picture. This function satisfies your condition but doesn't satisfy the task condition (here we see another counterexample).*

*Judgment: "It is not enough to eliminate zero value from the range! We must eliminate all negative numbers too! Therefore the power parameter must be even".*

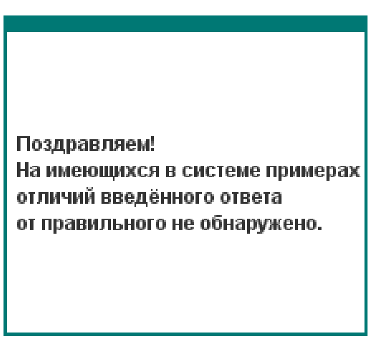



*Third trial*

*Input:* c>0 & m<0 & m_is_even

*Output. Your answer has a mistake. Look at the picture. This function satisfies task condition but doesn't satisfy your condition.*

*Judgment: "I forgot that the power parameter can be a fraction! We need to separate integer power parameters and fraction power parameters".*




*Fourth trial*

*Input:* c>0 & (m<0 & n>1 | n=1 & m<0 & m_is_even)

*Output. Congratulations! It is not possible to find difference between your answer and true one using accessible examples.*

**Using Verifier to support problem reformulation**

As M. Minsky noted [5], the reformulation of problem is one of the most important intellectual tools for its solving. For our purposes we will consider the process of reformulation as a process of knowledge representation forms alteration (Pozdnyakov, 1995, [9, 10]). For article's subject we will use four basic forms of knowledge representation:

- algorithmic representation (formulae)

- predicative representation

- representation by objects properties

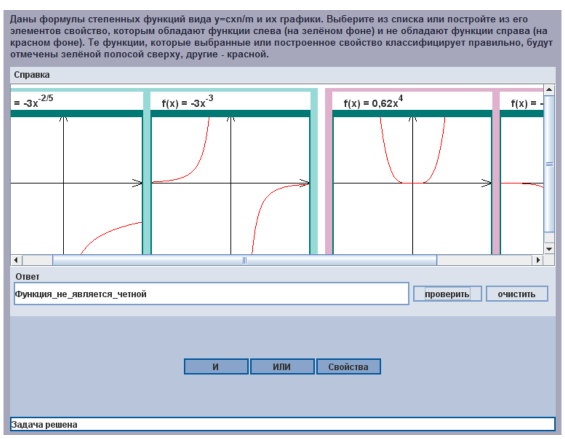
- representation by examples collection.

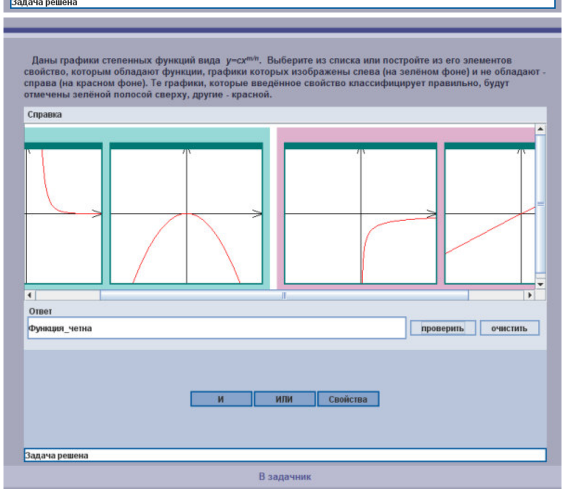
So there are many types of transitions from one form of representation to another.

The partial list of such transitions:

- from formula to graph
- from graph and formula to properties
- from the properties of a "class" object to a concrete example of this object with such properties
- from some set of properties to new one
- etc.

Beneath we will provide for analysis of some types of problems based on transitions from one form of knowledge representation to another.




*1. Type "from function graph and formula to function properties"*

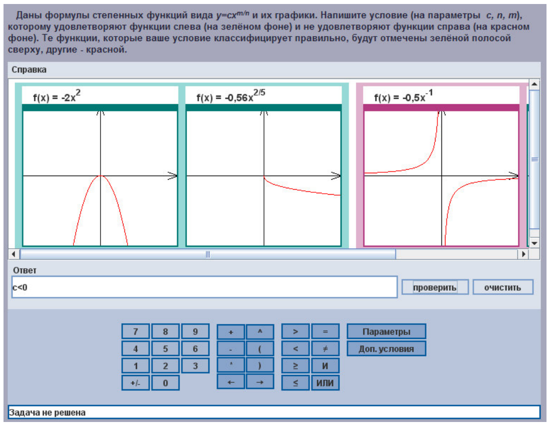
The type is characterized by giving the full information to a student about the object under consideration (function in our case). We give the set of examples and counterexamples of functions to a student with their formulae and graphs. The problem is to find an appropriate classification by formulating the property which separates examples and counterexamples.

*2. Type "from function graph to function properties"*

In this type of tasks we narrow the information about an object and provide the students with function graphs without any formulae.

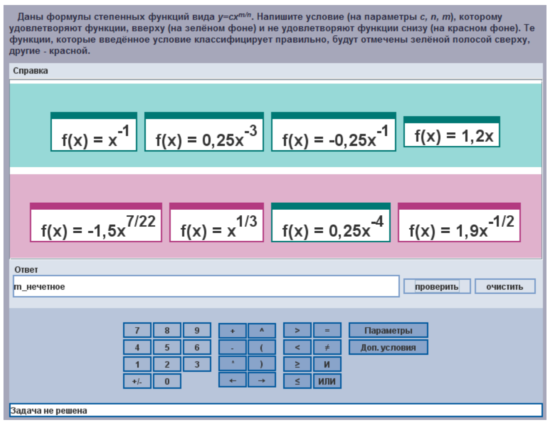
The problem for student is to find an appropriate classification by formulating the property to separate examples and counterexamples based on function graphs comparing.

Student must be aware of function graphs properties to succeed in solving this type of problems.



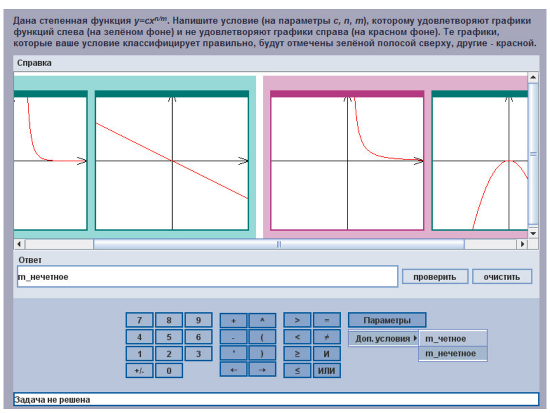

*3. Type "from function graph and formula to predicate"*

By stating these problems we may form the skills needed for solving inequalities. A student can search for a predicate (it is the same as to describe a solution by a set of inequalities) to specify a set of objects. Formulae here play the role of additional help in solution search.



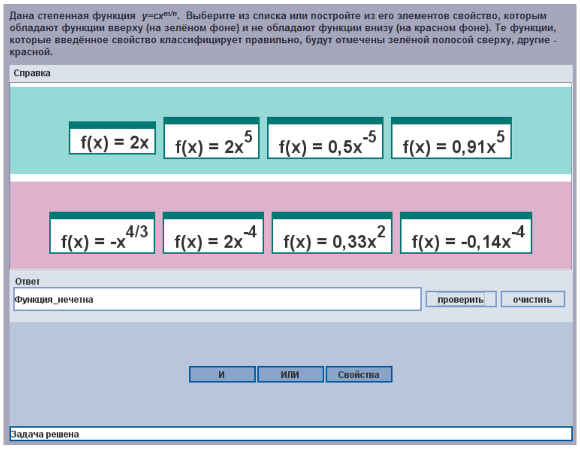

*4. Type "from function formula to predicate"*

Such type of the problems is characterized by function representation in a formula form. The search of solution requires a comparison between concrete formula and its class formula. This type of problems helps to train the skills of generating concrete examples for a common class and formulate characteristic properties of the object set.




*5. Type "from function graph to predicate"*

This is most important type of problems which combine common skills for information translation from one form to another with concrete math skills of exploring functions properties, solving equations and inequalities.

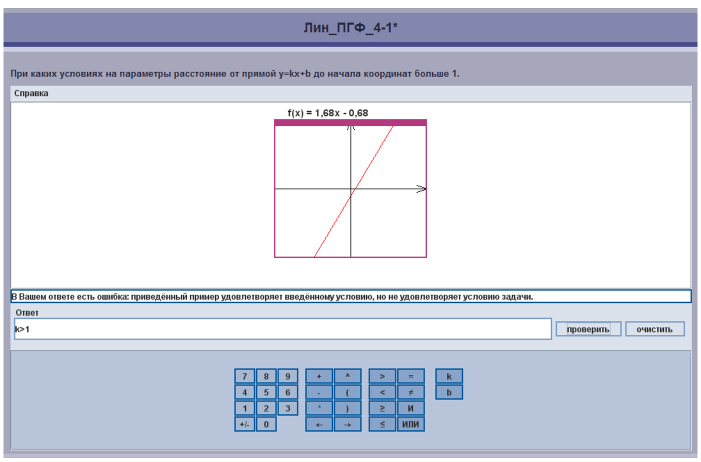



*6. Type "from function formula to function properties"*

The problems of such type are closely connected with translation of information from formulae language to verbal form. By such type problems we form math skills of object's "recognizing". For example: "what is an essential feature of function formula (from class given) to be even" or "what can be said about its domain.

**Further implications**

There are other types of problems, so here we provide their non-exhaustive list. They imply the examples construction in a symbolic or graph form.




*7. Type "from predicate to function graph".*

*8. Type "from predicate to function formula".*

*9. Type "from predicate to function graph and formula".*

*10. Type "from function properties to function graph and formula".*

# Research results and discussion

### Teachers’ answers on research questions

Beneath we cite some typical teachers’ answers which applied Verifier in their practice.

1. Is the material aimed to achieve new educational results?

"Yes. Students have learned to read attentively the problem description and to analyze it, noticing that problems, which seem to be identical, are different in fact".

2. Is it possible that Verifier usage in educational process have raised the level of student’s educational independence and self-activity?

"Yes, it is possible. The Verifier's problems put the students in a situation of self-control and self-estimations of actions".

3. Is it possible to state that using Verifier in educational process motivates students to learn math?

“Yes it is. The Verifier's problems are rather difficult and "time expensive", nevertheless students solved plenty of them, while working at home”.

4. What type of changes in real educational process was brought with the use of Verifier and were these changes necessary and effective?

“We didn’t feel any need in such changes”.

5. Does Verifier promote group and individual research activity in educational process?

“Yes. Verifier promotes different forms of group and individual research work”.

6. Does Verifier minimize labor expenses of the teachers when they prepare materials for lessons?

“Yes”.

7. Is Verifier suitable for schools, assuming it fits the hardware available at schools?

“Yes”.

8. What impact does Verifier have on quality of knowledge and skills?

“Students began to solve equations and inequalities with parameters more confidently. They also started to show awareness of function graphs properties”.

# Conclusion

The results of experiments with Verifier for supporting problems solving approve the authors’ position that verification environment can be used by a student as a tool to support his thinking about the complex calculus problems.

This conclusion is in good correspondence to works of L. Vygotsky in psychology about the role of tools in intellectual skills forming [11].

# References


1. M. Wertheimer. Productive Thinking.

2. Schoenfeld, A. (1985). Mathematical Problem Solving. New York: Academic Press.

3. G. Polya. How to Solve It.

4. G. Polya. Mathematical Discovery.

5. M. Minsky. Society of mind.

6. P. Winston. Artificial intelligence.

7. Sinclair, J., & Coulthard, M.: 1975, Towards an analysis of discourse, Oxford University Press, London.

8. Cazden, C.: 1988, Classroom Discourse: The Language of Teaching and Learning, Heinemann, Portsmouth, NH.

9. M. Bashmakov, S. Pozdnyakov, N.Reznik. Information environment of teaching process (rus.). “Svet”, Saint-Petersburg, Russia.

10. Pozdnyakov S. Mathematical models of intellectual dialogue in Computer Aided Learning. World Conference on Twenty-First Century Mathematics. Proseedings. School of Mathematical Sciences, GC University, Lahore, Punjab (Pakistan), 2005.

11. Vygotsky, L.S. (1962). Thought and Language. Cambridge, MA: MIT Press.