



OPEN ACCESS

EURASIA Journal of Mathematics Science and Technology Education
ISSN: 1305-8223 (online) 1305-8215 (print)
2017 13(8):5141-5155
DOI: 10.12973/eurasia.2017.00989a



Methodological Teaching System of Mathematical Foundations of Formal Languages as a Means of Fundamentalization of Education

Natalya A. Bushmeleva
Vyatka State University, RUSSIA

Tatiana A. Baklashova
Kazan (Volga region) Federal University, RUSSIA

Received 13 January 2017 • Revised 8 July 2017 • Accepted 10 July 2017

ABSTRACT

The purpose of the study is to develop content and identify appropriate methods, forms and means of studying formal languages that take into account the specific character of professional development of the future specialist. The methodological basis of the study is the system approach and the methodology for selecting the content of education, which allows to generalize and systematize the process of constructing the content of teaching formal languages, and to improve the content of training to specific formal languages. The article suggests a system for studying mathematical foundations of formal languages as a system-forming element in the training of specialists in the field of computer science, programming, and IT-technologies. It gives reasons for the inclusion of elements of the theory of formal languages in the learning process, defines the structure and content of learning tasks and objectives, develops the system of end-to-end concepts and the logical structure of the course program, suggests the order of presenting the training material, the system of training assignments for a laboratory course on a computer. The developed system allows graduates to get the opportunity to develop a high-level understanding of systems in general, which contributes to a common understanding of the structure of computer systems, and the processes of their creation and analysis.

Keywords: computer science and programming, fundamentalization, mathematical foundations, content of education, formal languages

INTRODUCTION

Economic and social transformations in Russia entail changes in many spheres of human life and activity, including education. It is reflected in its modernization. One of the directions of modernization is the fundamentalization of education. Specialists in computer science and programming need deep fundamental knowledge to help them develop new necessary skills as this area of science develops.

Fundamentalism of computer science education in the university can be achieved by combining theory, abstraction and implementation in the content of teaching theory, namely by studying the appropriate mathematical theories, algorithms and data structures, implementing algorithms and data structures in a specific programming language.

© **Authors.** Terms and conditions of Creative Commons Attribution 4.0 International (CC BY 4.0) apply.

Correspondence: Natalya A. Bushmeleva, *Department of Fundamental Informatics and Applied Mathematics, Vyatka State University, Kirov, Russia. Address to 36 Moscovskaya Street, Kirov City 610000, Russia. Tel: +7 9128287664.*

✉ na_bushmeleva@vyatsu.ru

State of the literature

- The development of the fundamental training system requires the development of teaching methods of its specific components - from choosing the content of training and searching for appropriate methods, forms and means of training, to forming the necessary student courses and introducing them into pedagogical practice.
- In the literature on the theory and methodology of teaching, the problems of teaching mathematical foundations of computer science, in particular, the teaching of the theory of formal languages, have been little studied. Existing textbooks on formal languages are often like unsystematic set of training texts. So it is time to generalize the collected information.
- There is not enough research on effective methods of teaching formal languages.
- At the moment, there is no classification of tasks in formal languages and, as a result, a «naive» approach to the selection of tasks is implemented in teaching.

Contribution of this paper to the literature

- For the first time, the structure of the content of teaching formal languages is clearly presented, as an ordered list of training elements combined into four blocks (introductory, syntactic, semantic, pragmatic) with strict internal organization. This is a structure, with clearly distinguished structure of the formal language, and its underlined semiotic nature.
- On the basis of the constructed general structure of the content of teaching formal languages, which allows to generalize and systematize the process of constructing the content of teaching formal languages. The content of training for specific formal languages has been adjusted.
- Methods of teaching formal languages are described. It allows teachers to use them in their activities. The variants of the thematic plans of the academic disciplines that provide training of students in formal languages are presented.

Back in the 70s of the 20th century, academician Ershov (1994) proposed a system of training specialists in the field of computer science, programming, and IT-technologies, in which the module «Fundamentals of programming» was the main one and included the following courses.

- 1) Introductory course of programming, which explains substantial bases of programming to freshmen.
- 2) «Mathematical foundations of programming» is a methodological course that reveals the essence of the mathematical method. The idea of the course is to teach the applied specialist to use his/her bundle of mathematical knowledge.
- 3) «Language bases of programming». Remarkably, this course should be taught outside of a specific language. That is, a student has to study and master the basic semantic concepts underlying the programming and learn how to use them, realizing the solution of the problem in a specific programming environment in a particular programming language.
- 4) The course «Technological fundamentals of programming» integrates all that one needs to know and be able to do, to come up, write, and debug one's program.
- 5) «System programming», involving the creation of large software products, software packages, information systems in a team or a creative group.

In international recommendations for the teaching of Computer Science discipline, these modules are also reflected in the following sections: Algorithms and Complexity (AL), Programming Languages (PL), Architecture and Organization (AR) (Computer Science Curricula, 2013).

Fundamentalization of education in the field of computer science, as in any other subject area, is achieved by including in the content of teaching ideological, philosophical and mathematical foundations of the academic subject. That is, elements relating to the theoretical foundations of computer science. One of such elements is mathematical foundations of computer science, in particular, the constructive part of discrete mathematics, integral

part of which is the formal language system. This will make it possible to teach formalization of domain theories with the help of formal languages (Laptev & Ryzhova, 2002).

The theory of formal languages forms the basis of syntactic methods (Hausser, 2014). Syntactic methods have wide practical application in the development and implementation of programming languages. In addition, artificial languages that use the text representation for describing the subject area are widely used not only in programming, but also in other areas. With their help, the structure of all kinds of documents, three-dimensional virtual worlds, graphical user interfaces, and many other objects used in models and in the real world is described. In order for these text descriptions to be correctly implemented, and then correctly recognized and interpreted, analysis and transformation methods are used. The theory of formal languages has given a push to the development of mathematical linguistics and methods of artificial intelligence associated with natural and artificial languages.

Therefore, with the development of the concept of fundamentalization of the subject training of graduates of IT-directions, qualitative fundamental training of students in the field of computer science is impossible without a serious study of some set of formal languages.

Fundamental mathematical knowledge for specialists in the field of IT-technologies and programming, as indicated by the international guidelines for the teaching of the disciplines Computer Science, include: the concept of «algorithm» and its refinement; the notion of «complexity of algorithms»; propositional calculus; first order predicate calculus; formal languages and ways of their describing, formal grammars; denotational and operational semantics; basic concepts of modal logic; basic concepts of the untyped λ -calculus; basic concepts of the calculus of interacting systems R. Milner (CCS calculus); Boolean functions, graph theory, Petri nets, schemes of programs, general and Boolean algebra; discrete structures; mathematical logic and formal semantics; formal languages and theory of automata; formal specification and verification; computability and computational complexity.

In teaching practice, there are several options for studying the issues above.

- 1) Introduction to the educational process of a new academic discipline «Mathematical Foundations of Computer Science».
- 2) Inclusion of the above issues in the content of the existing academic discipline «Theoretical Foundations of Computer Science».
- 3) Distribution of mathematical foundations of computer science as separate modules for the academic disciplines «Discrete Mathematics», «Mathematical Logic», «Theory of Algorithms», «Theoretical Foundations of Informatics», and «Formal Languages and Grammars».

LITERATURE REVIEW

The study of the problem of creating a methodological theory of teaching formal languages as a component of the system of fundamental training in the field of computer science shows that the accumulated experience in teaching formal languages is somewhat unsystematic, representing a lot of educational texts. A somewhat «naive» approach to the selection of practical problems is demonstrated in the educational and a few methodological literature (Bilgaeva, 2000; Ryzhova, 2000; Pentus & Pentus, 2004; Volkova, Vylitok & Rudenko, 2009), and there is no classification. In addition, researchers do not suggest effective methods of teaching formal languages. One of the few works on this topic is devoted to the method of teaching the formal languages of future computer science teachers (Fomina, 2003).

At the same time, it is the mathematical theory of formal languages that is the backbone of the above-mentioned disciplines. This approach allows the objective of teaching individual disciplines not to focus on individual concepts and topics, which subsequently does not lead to fragmentary mastery of the discipline.

In studies of problems of higher professional education, special attention is paid to fundamentalization issues (Kinelev, 1996; Shvetsky, 1996; Subetto, 1995; Beshenkov et al., 2016 and others). All authors note that the most important task of fully mastering the future profession and forming a professional outlook of the graduate of the university is to ensure the fundamentalization of education on the basis of systematic natural science knowledge that allows passing to a qualitatively new level of specialist training.

In the literature, there are many definitions of the concept of the fundamentalization of education. G. I. Kirillova (2001) characterizes the fundamentalization of information and computer training as a process aimed at ensuring the fundamental nature of this training; that is, the quality determined by the priority for its purposes and the composition of fundamental, long-lived, scientific information knowledge and generalized skills focused on the information component of professional activity. According to M. V. Bulanova-Toporkova (2002), the fundamentalization of education effectively contributes to the formation of creative engineering thinking, clear understanding of the place of their profession in the system of universal human knowledge and practice. N. V. Sadovnikov (2011) suggests training a specialist so that he himself could quickly adapt in a changing situation, giving him knowledge that is universal in nature, on the basis of which a specialist can quickly change himself in a new situation (Sadovnikov, 2011). The work of C. Chalmers et al. (2017) is devoted to the issue of creating integrated modules of the academic discipline in order to form students' deep knowledge of the discipline through the use of interdisciplinary ideas, which can be applied in other disciplines.

Scientists note that computer science, as a scientific discipline, has an exceptional multidimensionality. It can be regarded in the technological aspect - as a discipline close to computer science, as a natural science discipline. Such a point is held by N. N. Moiseev et al. (2001). Besides, it is a discipline of the language plan, which studies various formalized and formal languages. This opinion is held by V. K. Beloshapka et al. (1992).

Computer science is, in the opinion of many scientists, for example, A. A. Kuznetsov (2010), a complex scientific subject, which has an interdisciplinary nature, and today is called upon to perform integrative functions. The opinion of this scientist is confirmed by the study of E. P. Semenyuk (2016), in which computer science is considered in the context of the differentiation and integration of science. The study of V. P. Il'in (2015), A. G. Marchuk, T. I. Tikhonova & L. V. Gorodnyaya (2011) covers general problems of computational mathematics and computer science, and outlines ways of their further integration.

Various researchers have been involved in the issue of isolating the fundamental fields of knowledge for computer science: D. Knuth (1976, 1977, 1978), V. G. Kinelev (1996), M. Broy (1996), M. V. Shvetsky (1994), N. P. Ryzhova (2000), O. Chapman & S. An et al. (2017). Scientists, for example, K. Gravemeijer et al. (2017), in their studies try to answer the question: what kind of mathematical education can prepare students to the society of the future? The works of I. F. Yarullin, N. A. Bushmeleva & I. I. Trukhin (2015), R. A. Valeeva & N. A. Busmeleva et al. (2016) are devoted to the problems of forming mathematical competences of students. The article of N. J. Van der Wal, A. Bakker & P. Drijvers (2017), P. A. Fejer, A. Dan & D. A. Simovici (1991) identifies and justifies mathematical competences, which have to be formed in future professionals of technical profile.

As a result of their research, formal languages were called one of the most important components of the mathematical foundations of computer science. V. Myasnikov & N. Voskresenskaya (2005), K. Nakamura & I. Imada (2010) indicate that fundamentalization implies the inclusion in the curriculum of fundamental, basic concepts. A. A. Fomina (2003) notes that formal languages are the basic component of the mathematical foundations of computer science, which entails the need for a fundamental study of the course of the theory of formal languages (including a wide range of formal languages), and its categorical apparatus. The problems of forming students' fundamental concepts of the theory of formal languages are touched upon by many authors (Maróti, 2003; Armoni & Ben-Ari, 2009; Aleksandrov et al., 2013). It is important to note that today the categorical apparatus of computer science begins to penetrate into linguistics (Pobedonostseva & Shutikova, 2007; Pobedonostseva, 2016).

Analysis of methodological literature shows that, although the theory of formal languages is the basis in the construction of compilers, there is no generally accepted opinion on the question of what should be presented in the basic university course in this discipline (Karpov, 2012; Ryzhova, 2011).

Tasks and Objectives of the Study

The aim of the study is to build a methodological system for teaching the theory of formal languages as a component of the system of fundamental training of future specialists in the field of computer science, programming and IT-technologies. The following tasks were indicated as the main ones: to reveal the general structure of the content of teaching formal languages on the basis of the hierarchy of formal languages; to prove

the importance of formal languages in the system of fundamental training of future specialists in the field of computer science, programming and IT-technologies; to select and structure the methodological system of teaching formal languages; to allocate a general structure of the content of teaching formal languages on the basis of the analysis of existing training texts and the generalization of its results; to adjust educational content of specific formal languages on the basis of the allocated general structure; to identify the main classes of problems in formal languages and adjust the content of the task material; to describe the methods of teaching formal languages; to select effective experimental methods and to carry out an experimental test of some of the research provisions.

MATERIALS AND METHODS

Research Methods

The following methods were used to carry out the research: scientific and methodological analysis of the literature on philosophical, social and psychological-pedagogical problems related to the informatization of society, its impact on people and the education system; analysis of scientific literature on mathematics, computer science, computer technology, teaching methods of mathematics, and computer science; systematization and generalization of facts and concepts, analysis of educational standards, analysis of foreign and domestic programs for training specialists in the field of computer science, programming, and IT technologies, analysis of textbooks and teaching aids in formal languages and related disciplines; study, analysis and generalization of pedagogical experience in teaching the theory of formal languages; observation, analysis of the results of students' learning activities; method of expert estimations, forecasting, modeling, designing, development, and application of educational materials in education in the field of informatics, programming, and IT technologies, diagnostic techniques, pedagogical ascertaining and forming experiments on verification of individual theoretical positions and ideas.

Experimental Research Base

Approbation, generalization and implementation of research results are carried out:

- by teaching courses «Theoretical Foundations of Computer Science», «Theory of Automata and Formal Languages», «Formal Languages and Grammar» for students of various areas of training, implemented in Vyatka State University, in particular, Fundamental Computer Science and Information Technology, Applied Mathematics and Informatics, Pedagogical Education (mathematics, computer science) during 2010-2016;
- distance courses of professional retraining of personnel «Information Systems» (504 hours), in which teachers of profile subjects and school teachers took part;
- in the form of reports and speeches at scientific conferences and seminars of various levels, including international ones, publications in collections of scientific articles and scientific and methodical periodicals.

Stages of Research

The study is conducted in three stages.

During the first stage, the degree of elaboration of the research problem in domestic and foreign science was studied, philosophical, cultural, educational, psychological, technical and educational literature was analyzed, educational standards of higher professional education in the field of mathematical, technical skills and IT-technologies were studied and analyzed, their own experiences and experience of teachers were monitored and analyzed in order to explore the ways of fundamentalization of education in computer science, programming and IT-technologies and possible organizational forms of student activities to improve the effectiveness of the educational process.

At the second stage, approaches to the development of methodical system for teaching mathematical foundations of formal languages were studied, and their theoretical justification was based on the means of informatization. The content was selected for training sessions. Discussion of the practical implementation of the course model has been and continues to be implemented by feedback from students and listeners of the article of

the distance learning course conducted by the author, as well as during reports at conferences and seminars of various levels, which leads to consistent improvement of the proposed organizational model and methodology for its implementation in the educational process.

Within the third stage, the results, their qualitative and quantitative analysis, the preparation of a teaching aid and methodological recommendations for teachers were systematized and generalized.

RESULTS

Let us turn now to a direct description of the structure of the model, organizational forms of students' activities and some content aspects of studying the elements of the theory of formal languages in the professional training of graduates of higher education in the field of computer science, programming and IT technologies.

Analyzing the existing practice of teaching students computer science, programming in different areas of the physical, mathematical, technical, IT-profile, the concept of modern programming paradigms, we developed and substantiated the following theoretical questions that play a guiding role:

- 1) necessity and importance of teaching the theory of formal languages as a component of general vocational training;
- 2) selection criteria for educational-oriented programming languages to ensure the fundamentalization of the training of future specialists;
- 3) sequence of learning programming languages which support the most popular modern programming paradigms;

Then every component of the methodical training system was elaborated in detail:

- 1) structure and content of the learning tasks and objectives are defined;
- 2) technology of selecting the content of training is justified;
- 3) system of end-to-end concepts and logical structure of the curriculum of the discipline are developed;
- 4) structure and sequence of presentation of educational material, the system of educational and cognitive tasks are proposed;
- 5) methodology for teaching the theory of formal languages is created, and teaching and methodical manual for conducting laboratory course on computers is developed;
- 6) teaching theories of formal languages, methods of demonstration examples, conditional explanatory schemes, structural programming and computer modeling are theoretically substantiated and implemented in practice;
- 7) promising directions of development of the content are indicated.

The Concept of Fundamentalization of Education in the Field of Computer Science, Programming and IT-Technologies

The modern state educational paradigm in the context of the informatization of education calls for the need to shift the emphasis towards fundamental student preparation in the field of computer science, programming and IT technologies. This entails a change in the aims of education, among which the acquisition of fundamental interdisciplinary universal knowledge becomes principal. As a result, the main attention in the teaching of students should be given to general theoretical knowledge, with all their variety of internal and external links, forming the fundamental foundations of computer science, programming and IT-technologies. But at the moment, due to the rather large technological and applied focus of the subject area under consideration among scientists dealing with this problem, there is no common opinion about the fundamental foundations of educational disciplines related to the subject area of computer science, programming and IT technologies.

The fundamentalization of education presupposes, firstly, the allocation of a certain sufficient range of questions on the fundamental fields of knowledge in the direction of science and general educational disciplines, without which an intelligent person cannot exist; secondly, the study of this range of issues with full justification,

the necessary references, without logical gaps; thirdly, the identification of a set of basic laws and concepts that serves as a basis for the study of subsequent disciplines (Kuznetsova, 1995).

In addition, the fundamentalization of education should occur on the basis of mastering fundamental knowledge; that is, knowledge that combines the factual, philosophical and methodological scientific components of the subject. It should ensure the mastery of generalized activities that contribute to solving particular problems of the subject area.

Fundamentalization of education assumes that the content of training reveals not only the system knowledge of a particular subject area, but also a system of knowledge about the laws of mastering and the theoretical foundations for the search of ways to transfer the centuries-old experience of mankind (Stefanova & Shubina, 2001).

One way to improve the fundamentals in teaching computer science and programming is a reasonable combination in the content of teaching theory, abstraction and implementation. This means that in the process of preparing students, mathematical theories, algorithms and data structures, various implementations of algorithms and data structures in a particular programming language should be studied. This method should be supplemented by the inclusion of a certain formalization in the content of training, which is expressed in the construction of the language of a specific scientific discipline, since the fundamental aspect of both mathematics and computer science is the balance between theory and practice, their close connection with each other. With the described training approach, students, future specialists in the field of computer science and programming, get the opportunity to develop a high-level understanding of systems in general, which contributes to a common understanding of the structure of computer systems and the processes of their creation and analysis.

So, in the content of the academic subject, a considerable part of the learning elements is assigned to fundamental (theoretical) knowledge, which reveals the structure of the content of the subject area and activity reflecting the basic method of cognition. It is fundamental knowledge that reflects in the academic subject the grounds of the subject area, which include: the ideals and norms of knowledge of the domain, its philosophical foundations and the local picture of the world (Stepin, 2000). In addition, we should not forget about the philosophical and ideological foundations of the academic subject. The ideological, factual and methodological components of the fundamentalization of education form the basis of the academic subject.

Philosophical grounds are the basic philosophical concepts and directions, methods of cognition, characterizing the activity of researchers in a specific subject area.

World outlook bases are the abilities to use acquired knowledge about formalization (mathematical grounds) and knowledge of philosophical concepts (philosophical grounds) for solving problems in a specific subject area.

In this article, let us dwell in more detail on the mathematical foundations of the fundamentalization of education.

They provide students with knowledge of formalization, which is achieved through the construction of a formal language of the subject area (language of science, formal language, meta-language) and the formalization of existing content theories of this subject area. The connection between the substantive language of science and some artificially created formal language is described with the help of the meta-language. In this connection, the study of the theory of formal languages is extremely necessary in teaching the mathematical foundations of computer science and programming.

Elements of the Theory of Formal Languages in the Educational Process

The theory of formal languages forms the basis of syntactic methods. Syntactic methods have wide practical application in the development and implementation of programming languages. In addition, artificial languages that use the text representation for describing the subject area are widely used not only in programming, but also in other areas. With their help, the structure of all kinds of documents, three-dimensional virtual worlds,

graphical user interfaces, and many other objects used in models and in the real world are described. In order for these text descriptions to be correctly performed, and then correctly recognized and interpreted, analysis and transformation methods are used. The theory of formal languages has given a push to the development of mathematical linguistics and methods of artificial intelligence associated with natural and artificial languages. Therefore, with the development of the concept of fundamentalization, the subject training of graduates of IT-directions, qualitative fundamental training of students in the field of computer science is impossible without a serious study of some set of formal languages.

The formal language is a mathematical object of a high level of abstraction. It has a clearly distinguished formal structure, it is built on the basis of the separation of syntax and semantics, and it is the constructive object. The study of formal languages, as well as issues relating to the formalization of the family of languages, requires a high level of accuracy in presentation, deep knowledge in the field of abstract algebra. All of the above determines the peculiarities of the method of teaching formal languages as a subject and in the field of structuring the content of theoretical material, and when selecting material for practical work, and when choosing teaching methods.

Formal language is a mathematical model of a real language, that is, a way of communication of subjects with each other with the help of a finite set of signs (symbols) forming linear sequences (words). The communicative function of a language is studied using mathematical methods, which are based on the following principles.

- 1) Generalization (abstraction). Mathematical objects are formed by generalizing real objects. In the process of studying an object, only some of its properties are used, and the others are ignored.
- 2) The accuracy of reasoning. The correctness of logical conclusions is not verified experimentally, but the validity is shown to be accurate, subject to certain rules or reasons (evidence). It is the evidence that serves as the only way to justify the validity of a statement.

Thus, to learn the language using mathematical methods, firstly one needs to underline language properties that are important for study, and then – to strictly define these properties. The resulting abstraction will be called a formal language - a mathematical model of a real language.

In the process of developing a methodical system for teaching the formal languages of future professionals in the field of computer science and programming based on the concept of fundamentalization of education, taking into account the hierarchy of formal languages, the general structure of the content of teaching formal languages is identified on the basis of the analysis of existing teaching texts and generalization of its results. The content of the task material has been adjusted to specific representatives, and effective methods and means of teaching formal languages have been chosen.

The place of discipline in the structure of the basic educational program

The content of the discipline is integrative in its nature, using the achievements of a number of subjects: «Mathematical Logic and Theory of Algorithms», «Discrete Mathematics», «Mathematical Analysis», «Programming Basics», «Programming Languages», «Computer Practice», «Systemic and Application Software». «Theory of Formal Languages» is the basis for the study of «Software Engineering», «Intellectual Systems», and «Modeling Information Processes», for conducting research and development and for passing on-the-job training program.

Professional Competences Formed by the Discipline

- 1) The ability to understand and apply modern mathematical tools, fundamental concepts and system methodologies, international and professional standards in the field of information technology, the ability to use modern instrumental and computing tools in research and application activities.
- 2) Ability to professionally master basic mathematical knowledge and information technologies, effectively apply them to solve scientific and technical problems and applied problems associated with the development and use of information technology.
- 3) Understanding the concepts and abstractions, the ability to use the basic mathematical disciplines in practice.

As the result of studying the subject, the student has to:

- *know and understand* the relationship between languages and grammars; classification of languages and grammars; regular languages; deterministic and nondeterministic finite state machines; Turing machine; context-free grammars; ascending and descending analyzers; LL(k)-grammars and analyzers; LR(k)-grammars and analyzers; context-free grammars; context-dependent grammars; the use of grammars in modern compilers;
- *be able to* draw correct conclusions from the comparison of the results of theory and experiment; determine a class of grammars sufficient for the solution of the problem; build a grammar for the language; find the language generated by the grammar; build tables of LL(k)- and LR(k)-analyzers; use parser generators to build analyzers for grammar;
- *have* the skills of independent work in modern software systems; skills of mastering a large amount of information; skills of using syntax parsers.

Program of the Course «Elements of the Theory of Formal Languages»

Section 1. Mathematical foundations of computer science: formal languages.

Module 1. Preliminary information.

Module 2. Meta-languages.

2.1. Elements of the content theory of sets.

2.2. Elements of category theory. The concept of «category».

Module 3. Alphabetical languages.

3.1. Basic concepts: word, language, operation with languages in the alphabet. Classification of languages.

3.2. Calculus in words in the alphabet. Formal proofs in calculi in words in the alphabet. Associative calculi in the alphabet.

Module 4. Methods for assignment alphabetic languages in the alphabet.

4.1. Regular alphabetical languages. Regular expressions.

4.2. Formal grammar. Classification of formal grammars and languages. Classification of context-free grammars.

4.3. Conclusion and analysis task.

4.4. Algorithms for the transformation of context-free grammars.

4.5. Finite state machines as recognizers.

4.6. Finite state machines as converters. Elements of the algebraic automata theory.

Module 5. Combinatorial logic languages.

5.1. Elements of λ -calculus.

5.2. Elements of combinatorial calculus.

Module 6. First-order logical languages.

6.1. Syntax and semantics of the language. First-order language syntax.

6.2. Hilbert calculus of the first order. Construction of evidence and conclusions using the basic rules of inference.

6.3. Hilbert calculus of the first order. Establishment of the existence of evidence and conclusions by means of admissible inference rules.

Module 7. Logical and mathematical languages.

7.1. Formal first-order arithmetic (S theory).

7.2. Formal second-order arithmetic (S2 theory).

7.3. Formal Cantor (naive) set theory (M+).

Section 2. Mathematical foundations of computer science: formal syntax and semantics of programming languages.

Module 8. Formal syntax of programming languages.

8.1. Formal syntax of model languages.

Module 9. Operational semantics of programming languages.

9.1. The substantive (intuitive) operational semantics of the model languages D and H.

9.2. Machine with unlimited registers.

Module 10. Deductive semantics of programming languages.

10.1. Deductive semantics of the H language (Hoare system).

10.2. Wp predicate converter.

10.3. Simple statements. Assignment statement. Statement composition.

10.4. Dijkstra selection statement.

10.5. Dijkstra cycle statement.

10.6. Procedures in the Dijkstra system.

10.7. Synthesis of assignment statements. Synthesis of selection statements. Synthesis of cycle statements.

10.8. Construction of invariants.

10.9. Presentation of functions of the programming language M in the form of λ -terms and CL-terms. Proof of the correctness of functional programs.

Module 11. Denotational semantics of programming languages.

11.1. Denotational semantics of the model language.

Module 12. Elements of computer logic.

12.1. Algorithms for recognizing the type of zero-order language formulas.

12.2. Algorithms for recognizing the type of first-order language formulas.

12.3. Semantic resolution. Lock resolution.

12.4. Linear resolution.

Methodical Remarks

The sequence of the course topics was obtained as a result of the optimization of the curriculum of the course in content and in time. The content of the course consists of two parts. The first part is devoted to formal languages. The second – to the formal syntax and semantics of programming languages. Each of them consists of a set of modules. The module is a set of course topics that can either be moved, or taken out for independent study by students, or removed from the content of training, if students have already studied the concepts of this module during other courses.

Topics of the course as a part of the module are separate units of educational material that reveal the logical organization and interrelation of questions from the positions of mathematics and computer science, the accuracy level of the facts in question; and also allows to identify and formulate the goals of studying the basic concepts, to select possible most effective means and methods of teaching these concepts, to think through the system of monitoring and evaluation of the system of knowledge and skills.

The logical structure of the educational content of the mathematical foundations of computer science can be different. It can be changed depending on various factors (basic preparation of students, place in the basic educational program, the number of study hours) within the real educational process.

Thus, relying on the suggested model of the content of teaching the mathematical foundations of computer science, it is possible to build various versions of the academic discipline by choosing a number of topics depending on the actual conditions of the learning process.

The main types of classes in the study of discipline are lectures and practical classes, as well as independent work of students. During the study of the discipline attention is paid both to the theoretical assimilation of the basic concepts of the course of the theory of formal languages, and to the acquisition, development and consolidation of competencies, practical skills and skills in assigning the language, constructing finite automata for recognizing different languages, and applying the knowledge gained in solving practical problems. Lectures are conducted using multimedia and interactive technologies (group discussion of the field of application of methods of

development and analysis of algorithms in the context of specific tasks solved by the teacher and student, individual student consultations in the process of solving learning tasks in the computer class, individual student consultations through telecommunication technologies).

Practical classes are mainly focused on the study of set-theoretic operations on languages, the study of properties of operations over languages and methods for recognizing a language by an automaton. In the course of practical work, teaching materials of various types are used: electronic and paper textbooks, electronic collection of typical assignments, presentations, intelligence maps, etc. The results obtained in the course of practical work are talked over both individually and in a discussion format (interactive conferences).

Independent work of students is aimed at consolidating knowledge and skills obtained during lectures and practical classes. The results of independent work of students are necessarily controlled by the degree of mastering by students of the basic methods of working with algorithms.

Studying with elements of the theory of formal languages requires mastering the appropriate mathematical apparatus, methods of constructing finite automata, and specifying languages using grammars and regular expressions. Therefore, practical exercises have a fundamental role in teaching.

To successfully master these methods, it is necessary to:

- carefully select the content of practical exercises, giving preference to those tasks where the essence of the method of working with algorithms demonstrates its importance in the educational process;
- consistently implement the principle of «from simple to complex», i.e. when the method is mastered, it is necessary to acquaint students with more complex algorithms and methods for their analysis;
- carry out tasks and projects only with a professional orientation.

To implement the principle of individual approach in the classroom, students are offered individual tasks on options, author's approaches to their implementation are considered and approved. Brainy students perform tasks of increased difficulty. Students perform tasks independently, but this is preceded by a collective discussion of algorithms. The activity and content of students' presentations at interactive conferences are encouraged. The most interesting variants of solving problems are considered and analyzed together; projects are demonstrated in the form of presentations.

The organization of independent work within the framework of discipline study can be carried out taking into account the three levels of students' activity: the reproductive (training) level, the reconstructive level and the creative (search) level. Training independent work is carried out on the model: tasks solution is carried out according to well-known algorithm. The student's cognitive activity manifests itself in recognition, comprehension, and remembering. The purpose of this kind of work is to consolidate knowledge and form skills. In the course of performing reconstructive independent work, decisions are being made; a new approach to the solution of the problem is made with a combination of methods and technology solutions. This kind of independent work contributes to the development of reflection skills. Creative independent work requires an analysis of the problem situation, obtaining new information. The student must independently choose the means and methods of solution (creative projects, study and research assignments). Independent work is carried out using a wide range of interactive technologies. The share of classes where interactive technologies are applied is 54%. As the hardware, multimedia equipment, network technologies, and interactive boards are used, which ensure efficient work with various information sources. This allows one to implement both individual learning paths and collective interactive forms of learning. The main technology for working in the classroom is individual work in the Maple environment, which allows students to visualize the course of solving the problems of constructing finite automata, assigning languages, simplifying the search for mistakes and contradictions. In the classroom, modern technologies related to the class of MindMap technologies are widely used. To activate the thinking activity of students, develop memory, the ability to structure information, effectively take notes and annotate the material of lectures, the FreeMindMap version 0.9.0 is used. These programs allow students to synchronize the process of reasoning with visualization, which makes the reasoning plan accessible to the audience. Any student can express his or her position in the general plan of reasoning with fixation.

Didactic Support of the Course «Elements of the Theory of Formal Languages»

Didactic support of the course "Elements of the theory of formal languages" is a system of exercises and tasks on the mathematical foundations of computer science, which helps to consolidate the content of the main theoretical material. The developed system contains four types of tasks in accordance with the classification of tasks related to the topic of formal languages:

1. tasks related to formal systems and their interpretation in accordance with their classification;
2. tasks of studying the structure of a formal language, identifying its constructive part (proof of existence theorems, algorithms from the formal theory);
3. tasks to study the complexity of the algorithms obtained;
4. tasks related to the formal syntax and formal semantics of programming languages.

DISCUSSIONS

The theory of formal languages is, in a certain sense, the mathematical basis of IT development, it can form an integral part of the general culture of an IT specialist skillfully using modern information, computer, multimedia, and telecommunication products. The teaching of the theory of formal languages should be directed to the appropriate training of the future professional training methods of computer simulation and the formation of components of the mathematical style of thinking of the future professional. V. Devedzic & J. Debenham (1998) described an intellectual learning system for teaching formal languages.

Analysis of problem development of systematic teaching of the theory of formal languages has not been largely solved for students wishing to associate their professional activities with computer science, programming and IT-technologies.

The developed course «Elements of the theory of formal languages» is a system of teaching mathematical foundations of formal languages and fundamentalization of education in the field of computer science, programming and IT-technologies, because it is based on the study of the existing semantics of fundamental concepts. The possibility of its mastering by the students is in many respects connected with the constructive embodiment of concepts in the computer, which makes it possible to ensure the practical activity of the students in the process of studying the concepts of theoretical computer science.

CONCLUSION AND RECOMMENDATION

This research, having confirmed the need to study a certain set of formal languages to ensure the qualitative fundamental training of future specialists in the field of computer science, programming and IT-technologies, made it possible to construct a methodological system for teaching formal languages and to consider it as a means of fundamentalization of education. It is based on the state educational paradigm, which focuses on the fundamental character and integrity of students' knowledge.

During the research, incorporation of elements of theory of formal languages as the component of general professional training in the learning process is justified; the structure and content of the goals and objectives of the training are defined. Then, on the basis of the chosen content selection technology, the system of end-to-end concepts and logical structure of the course program were developed, the sequence of presentation of training material and the system of study assignments were proposed, the training manual for conducting the laboratory course on computers was developed. In addition, the methods of demonstration examples, conditional explanatory schemes, structural programming, and computer modeling were theoretically proved and introduced into practice of teaching; promising areas of content development were defined.

As the interpretation of the developed methodical system, there are two possible options: the first is designed for a separate training course of formal languages and following directly from the ideas of the concept of fundamental training; the second is based on the reality of the educational process at the Faculty of Computer and Physical and Mathematical Sciences of Vyatka State University. Thematic planning of formal languages for

postgraduate students of the Department of Fundamental Informatics and Applied Mathematics of Vyatka State University was also suggested.

The materials of the article can be useful in training students in the field of computer science, programming and IT-technologies. Teachers on the basis of the constructed general structure of the content of teaching formal languages can adjust the content of instruction to specific formal languages using the methods of teaching formal languages described in the article, which will increase the effectiveness of the educational process.

We see the promising direction of improving the proposed methodological system in its complementing with aspects related to the wide introduction of technologies for distance learning, namely: remote support of student activities during the educational period. In addition, it is possible and necessary to create training content in various formal languages; further elaboration and development of classification of tasks on formal languages and other similar fields of knowledge; study of the learning process of formal mathematical theories; search and selection of general learning content structures for other classes of objects similar to formal languages (for example, various abstract structures).

ACKNOWLEDGMENT

This work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

REFERENCES

- Aleksandrov, A. V., Kazakov, S. V., & Sergushichev, A. A. (2013). The use of evolutionary programming based on training examples for the generation of finite state machines for controlling objects with complex behavior. *Journal of Computer and Systems Sciences International*, 52, 410-425. doi:10.1134/S1064230713020020
- Armoni, M., & Ben-Ari, M. (2009). The Concept of Nondeterminism: Its Development and Implications for Teaching. *Science & Education*, 18(8), 1005-1030. doi:10.1007/s11191-008-9147-5
- Beloshapka, V. K. (1992). Computer science as a science of letters. *Informatics and Education*, 1, 6-12.
- Beshenkov, S. A., Mindzaeva, E. V., Beshenkova, E. V., Shutikova, M. I., & Trubina I. I. (2016). Information Education in Russia. In V. Uskov, R. Howlett, & L. Jain, (eds.), *Smart Education and e-Learning 2016. Smart Innovation, Systems and Technologies*, 59, 563-571. doi:10.1007/978-3-319-39690-3_50
- Bilgaeva, N.Ts. (2000). *Theory of algorithms, formal languages, grammars and automata: Teaching guide*. Ulan-Ude: ESSUTM.
- Broy, M. (1996). *Informatics. Fundamental introduction. In 4 pt. 4.1*. Moscow: Dialogue-MEPH.
- Bulanova-Toporkova, M. V. (2002). *Pedagogy and psychology of the higher school*. Rostov-on-Don: Phoenix.
- Chalmers, C., Carter, M., & Cooper, T. (2017). Implementing "Big Ideas" to Advance the Teaching and Learning of Science, Technology, Engineering, and Mathematics (STEM) *International Journal of Science and Mathematics Education*, 15(Suppl 1), 25-43. doi:10.1007/s10763-017-9799-1
- Chapman, O., & An, S. (2017). A survey of university-based programs that support in-service and pre-service mathematics teachers' change. *ZDM Mathematics Education*, 49(2), 171-185. doi:10.1007/s11858-017-0852-x
- Computer Science Curricula (2013). *Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. New York, NY, USA: ACM.
- Devedzic, V., & Debenham, J. (1998). An Intelligent Tutoring System for Teaching Formal Languages. In B. P. Goettl, H. M. Halff, C. L. Redfield, & V. J. Shute, (eds.), *Intelligent Tutoring Systems. ITS 1998. Lecture Notes in Computer Science*, 1452, 514-523. Springer, Berlin, Heidelberg. doi:10.1007/3-540-68716-5_57
- Ershov, A. P. (1994). *Selected works*. Novosibirsk: Nauka.
- Fejer, P. A., Dan, A. & Simovici, D. A. (1991). *Mathematical Foundations of Computer Science. Sets, Relations, and Induction*, 1. doi:10.1007/978-1-4612-3086-1
- Fomina, A. A. (2003). *Method of training future teachers of computer science in formal languages* (PhD Thesis). Russian State Pedagogical University named after Herzen, St. Petersburg.

- Gravemeijer, K., Stephan, M. & Julie, C. (2017). What Mathematics Education May Prepare Students for the Society of the Future? *International Journal of Science and Mathematics Education*, 15(Suppl 1), 105-123. doi:10.1007/s10763-017-9814-6
- Hausser, R. (2014). *Foundations of Computational Linguistics*. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-41431-2
- Il'in, V. P. (2015). Computational mathematics and informatics: Global challenges and Russia's roadmap. *Herald of the Russian Academy of Sciences*, 85(1), 8-14. doi:10.1134/S1019331615010098
- Karpov, Y. G. (2012). *Theory and technology of programming. Fundamentals of building compilers*. Moscow: BHV.
- Kinelev, V. G. (1996). Education and civilization. *Informatics and Education*, 5, 21-28.
- Kirillova, G. I. (2001). *Optimization of the content of information and computer training in the secondary vocational school* (PhD Thesis). Kazan State Pedagogical institute, Kazan.
- Knuth, D. (1976). *Art of computer programming. V.1: Basic algorithms*. Moscow: Mir publ.
- Knuth, D. (1977). *Art of computer programming. V.2: Seminumerical algorithms*. Moscow: Mir publ.
- Knuth, D. (1978). *Art of computer programming. V.3: Sorting and searching*. Moscow: Mir publ.
- Kuznetsov, A. A. (2010). *Fundamentals of general theory and methods of teaching computer science: Teaching guide*. Moscow: BINOM. Knowledge Laboratory.
- Kuznetsova, V. A. (1995). *Theory and practice of multilevel university pedagogical education*. Yaroslavl: YarSU.
- Laptev, V. V., & Ryzhova, N. I. (2002). The concept of fundamentalization of education in the field of computer science and its implementation in a pedagogical university. *Izvestia: Herzen University Journal of Humanities & Sciences*, 3, 124-135.
- Marchuk, A. G., Tikhonova, T. I., & Gorodnyaya, L. V. (2011). Novosibirsk Young Programmers' School: A Way to Success and Future Development. In J. Impagliazzo, & E. Proydakov, (eds.), *Perspectives on Soviet and Russian Computing*, 228-234. *IFIP Advances in Information and Communication Technology*, 357. Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-22816-2_27
- Maróti, G. (2003). Didactic approach for teaching nondeterminism in automata theory. *Zentralblatt für Didaktik der Mathematik*, 35(2), 48-55. doi:10.1007/BF02652772
- Moiseev, N. N. (2001). *Universe. Information. Society*. Moscow: Ustoichivyi mir.
- Moshchensky, A. V., & Moshchensky, V. A. (2008). *Mathematical foundations of computer science*. Minsk: BSU.
- Myasnikov, V., & Voskresenskaya, N. (2005). Compare, study, critically evaluate. *Uchitel*, 2, 6-10.
- Nakamura, K., & Imada, K. (2010). Incremental Learning of Cellular Automata for Parallel Recognition of Formal Languages. *International Conference on Discovery Science*, 117-131. doi: 10.1007/978-3-642-16184-1_9
- Pentus, A. E., & Pentus, M. P. (2004). *Theory of formal languages*. Moscow: Publishing house of MSU Centralized Institute of Mechanics and Mathematics.
- Pobedonostseva, M. G. (2016). Requirements for education at the present stage of society. In collection: Actual problems of the methods of teaching computer science in a modern school. *International scientific and practical Internet conference*.
- Pobedonostseva, M. G., & Shutikova, M. I. (2007). Intersubject communications of computer science. *TSU Journal, Tambov: Derzhavin Tambov State University*, 12(5), 621-622.
- Ryzhova, N. I. (2000). *Development of the methodical system of fundamental training of future computer science teachers in the subject area* (PhD Thesis). St. Petersburg: RSEU Publishing House.
- Ryzhova, N. I. (2000). *Elements of theoretical computer science: Exercises on the mathematical foundations of computer science: formal semantics of programming languages: Teaching guide for students*. St. Petersburg: RSEU Publishing House.
- Ryzhova, N. I. (2011). Mathematical foundations of computer science as an element of mathematical training of computer science teacher. *World of Science, Culture & Education*, 5, 158-163.
- Sadovnikov, N. V. (2011). Fundamentalization of contemporary education. *Izv. Penz. gos. pedagog. univ. im.i V. G. Belinskogo*, 24, 782-786.

- Semenyuk, E. P. (2016). Informatics in the context of the differentiation and integration of science. *Scientific and Technical Information Processing*, 43(1), 8-19. doi:10.3103/S0147688216010044
- Shvetsky, M. V. (1994). *Methodical system of fundamental training of future teachers of computer science in a pedagogical university in conditions of two-level education* (PhD Thesis). St. Petersburg: Russian Pedagogical Institute.
- Stefanova, N. L., & Shubina, N. L. (2001). Fundamental science and fundamental education: the problem of correlation and interaction. *Academic readings*, 2, 27-30.
- Stepin, V. S. (2000). *Theoretical knowledge*. Moscow: Progress-Traditsiya.
- Subetto, A. I. (1995). *Problems of fundamentalization and sources of the content of higher education: Facets of public policy*. Kostroma: Kostroma ped. un.
- Valeeva, R. A., & Bushmeleva, N. A. (2016). Forming analytical competency of higher school students. *IEJME-Mathematics Education*, 11(8), 3137-3148. doi:10.12973/mathedu.2015.109a
- Van der Wal, N. J., Bakker, A., & Drijvers, P. (2017). Which Techno-mathematical Literacies Are Essential for Future Engineers? *International Journal of Science and Mathematics Education*, 15(Suppl 1), 87. doi:10.1007/s10763-017-9810-x
- Volkova, I. A., Vylitok, A. A., & Rudenko, T. V. (2009). *Formal grammars and languages. Elements of translation theory*. Moscow: Publishing Department of the Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University.
- Yarullin, I. F., Bushmeleva, N. A., & Tsyrukun, I. I. (2015). The research competence development of students trained in mathematical direction. *IEJME-Mathematics Education*, 10(3), 137-146. doi:10.12973/mathedu.2015.109a

<http://www.ejmste.com>