

resources, including study materials, assignments, and tests;

- the ability to receive feedback from the system and teachers to assess progress and correct the learning process.

The main disadvantages of an intelligent tutoring system:

- limited ability of the system to adapt to the specific needs of students with different backgrounds and learning abilities;
- lack of accuracy in identifying student errors and failures, which may lead to underestimation of their real abilities;
- necessity of constant updating of educational content and algorithms of the system in accordance with changes in the discipline program and teaching methods;
- limited opportunities to interact with students outside the learning environment, which can make it difficult to solve individual questions and problems.

One of the best known systems in problem solving is **WolframAlpha** [2]. This system can solve problems from various fields, including graph theory problems.

This system supports functions such as graph editing, basic graph operations, providing examples with known graphs, and searching for elements in a subgraph.

However, the disadvantages of WolframAlpha are that it cannot provide step-by-step solutions to problems, it has a special query language that not every user can understand, and it does not provide training tasks for users.

ALEKS is an intelligent tutoring system that provides a customizable program for teaching students in different subject areas, including discrete mathematics [3]. The system uses an approach based on passive learning theory, which allows hypothesizing about a student's knowledge and determining the most effective learning path based on the student's learning data.

The ALEKS system uses mathematical algorithms that are the basis of its knowledge base and artificial intelligence technology. The ALEKS knowledge base is a system of concepts and tasks that are interconnected and divided into levels of complexity. Each assignment has its own unique difficulty level, which is determined based on the student's study data. This allows students to learn at their own pace and make the most efficient use of their time.

The disadvantages of this approach are its limited depth of understanding of topics and its inability to integrate with other systems. This can lead to limitations in using the system within broader programs of study and limitations in accessing deeper knowledge of discrete mathematics. In addition, a customizable

curriculum may be less effective with students who already have a certain level of knowledge in discrete mathematics and do not require such customization.

Webwork is a system of online teaching materials [4]. This system allows teachers to create individual assignments and exercises that students can solve in real time, and also provides the ability to automatically create different versions of assignments and assessments, and use interactive elements such as graphs and visualizations to facilitate the understanding of the material.

The Webwork information base consists of mathematical formulas, algorithms for solving problems, and a database containing assignments and students' answers. The Webwork approach allows new assignments and answers to be quickly added to the database, which is its advantage. However, the disadvantage is the limited functionality, since Webwork cannot use complex mathematical objects such as graphs or matrices. Also, due to the lack of systematization and semantic structuring of knowledge, it is impossible to automate the process of creating tasks and generating answers, and in the absence of a certain problem in the database its solution will not be found.

Maple T. A. is a program that allows teachers to create and automatically check tests and assignments in discrete mathematics [5]. Maple T.A. uses a combination of several approaches. First, this system uses traditional mathematical methods to develop an information base, including the creation of test assignments and courses, and a semantic approach to create knowledge models and intelligent data processing systems. Second, it utilizes machine learning algorithms for adaptive learning and assessment of student knowledge, and incorporates automatic answer checking, manual evaluation, and proof checking. It is possible to add new mathematical objects such as graphs or matrices, which is an advantage of the system.

The system has limited functionality and no integration with other systems, which may limit its use in some areas. For example, in a business domain where it is important to have access to different systems and tools to solve problems, this limitation can be a significant disadvantage. In addition, the limited functionality may not meet the needs of users, which can lead to loss of interest and unproductive use of the system.

An intelligent tutoring system for discrete mathematics should not have the disadvantages inherent in the above systems, namely it should provide the ability to view step-by-step solutions, ease of integration with other systems, and ease of integrating new skills, knowledge, and themes into the existing system.

Also, the above systems are only able to solve a limited range of tasks that are predetermined by the developers, and the systems cannot generate tasks themselves for the user training.

II. Proposed approach

The proposed approach implies the developing of a system based on the *OSTIS Technology* and its basic principles to overcome these disadvantages [6].

OSTIS Technology is a complex *intelligent system* design technology based on semantic knowledge representation, which includes:

- library of generic, reusable and semantically compatible components and *intelligent systems* (components of *knowledge bases*, *intelligent problem solvers*, *intelligent user interfaces*);
- compatible semantic knowledge representation languages of various kinds, providing semantic compatibility not only for reusable components of *intelligent systems*, but also for entire *intelligent systems*;
- compatible semantic models of problem solving.

As a formal basis for knowledge representation within the framework of *OSTIS technology*, a unified *semantic network* with a set theory interpretation is used. Such a representation model is called *SC-code* (Semantic computer code). The elements of such a *semantic network* are called *sc-nodes* and *sc-connectors* (*sc-arcs*, *sc-edges*) [7]. A model of an entity described by means of *SC-code* is called a *sc-model*.

Intelligent systems developed with the use of the *OSTIS Technology* are called *ostis-systems*. Each *ostis-system* consists of a platform-independent unified logical-semantic model of this system (*sc-model* of a computer system) and a platform for interpretation of such models. In turn, each *sc-model* of a computer system can be decomposed into *sc-model* of a *knowledge base*, *sc-model* of a *knowledge processing machine*, *sc-model* of an *interface* and an *abstract sc-memory* where *SC-code* constructions are stored [7].

The creation of *sc-models* of the *knowledge base* is based on the ontological approach, which implies the creation of ontologies as systems of concepts describing a particular subject area [8].

The problem solver is the main part of the *sc-model* of the *knowledge processing machine*, which is built on the basis of a *multi-agent* approach, where the interaction of *agents*, called *sc-agents*, is carried out exclusively by means of *semantic memory*, which stores *SC-code* constructions [9]. This approach allows to ensure modularity and flexibility of the developed machine, and also provides the possibility of parallel execution of different knowledge processing processes.

III. Knowledge base

An important step in the design of *intelligent tutoring systems for discrete mathematics* is the creation of a *knowledge base* that will contain complete and structured information on *discrete mathematics*.

However, in order for this *knowledge base* to be as useful and effective as possible, it is necessary to consider a number of criteria that will allow it to be evaluated.

Based on this, the following basic and most important criteria for analyzing a knowledge base were highlighted:

- *Knowledge base for discrete mathematics* can be organized as a tree structure, where each node represents a particular topic, or as a network structure, where each node represents a different theorem or algorithm. Evaluating the structure of the *knowledge base* may include analyzing the hierarchy of topics, the relationships between topics and individual elements of the *knowledge base*, and the ease of navigating the *knowledge base*.
- *knowledge base for discrete mathematics* should contain sufficient information about key concepts, theorems, algorithms, and applications. Assessment of the quality and completeness of the information may include verifying that all necessary definitions and theorems are present and that the information provided is accurate and reliable.
- *knowledge base for discrete mathematics* should be user-friendly and easily accessible for use by students, teachers and researchers. Evaluation of functionality and usability may include an analysis of the accessibility of the *knowledge base* and the ability to search for information.

An important factor affecting the efficiency and quality of the system is the structured nature of the *knowledge base*. For this purpose, it is necessary to have an organized hierarchy of partial *subject domains*, which are strongly connected with each other by many different relations related to some common (base) *subject domain*.

Figure 1 contains an example of the description of a **subject domain of graph theory**.

Determining the relations between concepts is an important step in the development of a *knowledge base*, and requires careful analysis and careful approach. According to *OSTIS Technology*, to achieve the best quality of the *knowledge base*, concepts and relations will be described using their *semantic neighborhoods*. The basic relations and concepts include identifier, definition, statement, inclusion and decomposition. The implementation of each relation and their representation in the *knowledge base* depends on the particular case.

The next step is to define the rules and principles of describing the elements of the *knowledge base*. If there are several concepts in the *knowledge base* that are related to each other, the correct description of these concepts and connections between them allows to avoid duplication of information and ensure the integrity of the *knowledge base*.

Figure 2 shows an example of the description of the concept **disconnected graph**.

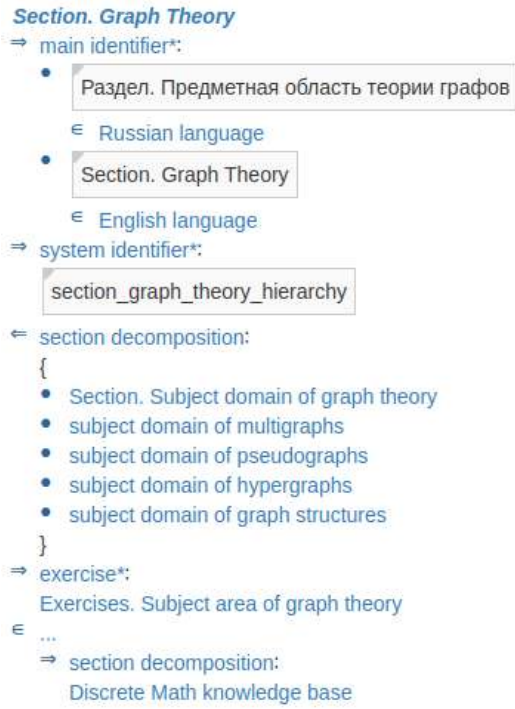


Figure 1. Section. Subject domain of graph theory

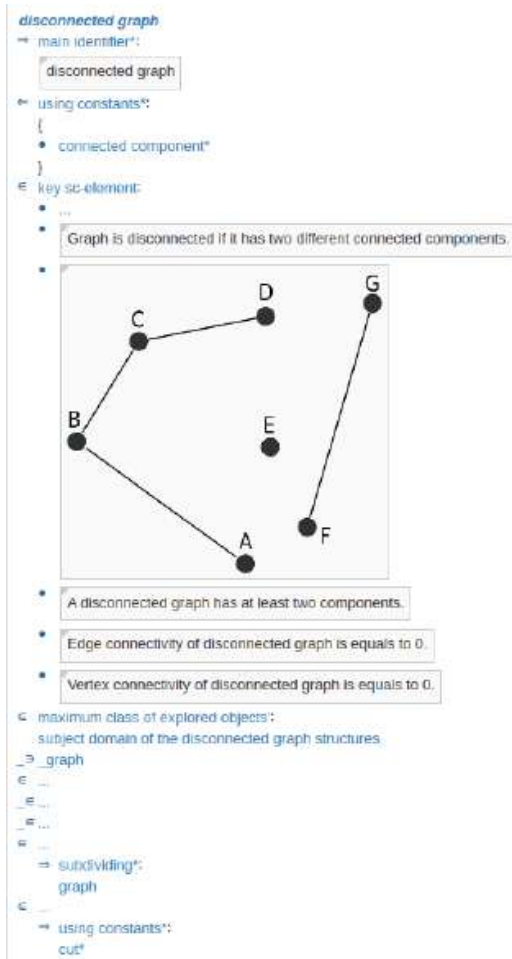


Figure 2. Concept disconnected graph

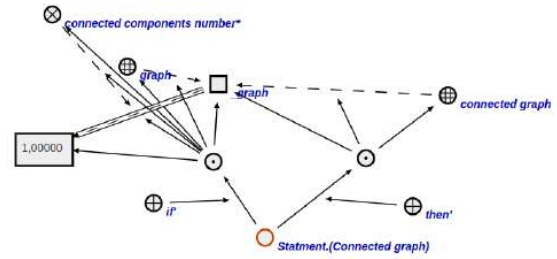


Figure 3. Statement about connected graph

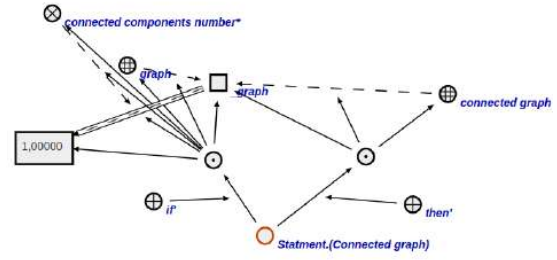


Figure 4. Specification of program for finding the union of two graphs

A unified description of the elements of the *knowledge base* creates an opportunity for its expansion in the future. The predefined rules and principles for describing elements will allow us to quickly and easily add new elements to the *knowledge base* without having to redesign its entire structure. Thus, we can be sure that system is flexible and scalable [10].

An example of a statement about a *connected graph* is shown in Figure 3, it is a logical formula written as an implication. The premise of the implication contains a pattern denoting that the graph has only one connectivity component. In the conclusion of the implication, the graph belongs to the set of connected graphs.

The *knowledge base* also contains specifications of programs that can be used by the *problem solver*. Thus, Figure 4 shows the specification of the *graph union finding program*. The result of this program is a graph that is the union of two input graphs.

An example of an exercise is shown on Figure 5.

IV. Problem solver

An important step in creating a *intelligent tutoring system for discrete mathematics* is to develop a *problem solver* that will solve problems in *discrete mathematics* with a full description of the step-by-step solution. But in order for the *problem solver* to fully fulfill its function, it is necessary that the *problem solver* meets the following criteria: