Modeling set theory laws using maple computer algebra system

K T Tyncherov¹, A A Olenev², M V Selivanova¹, K A Kirichek²
¹ Ufa State Petroleum Technological University, Branch of the University in the City of Oktyabrsky, 54a, Devonskaya St., Oktyabrsky, Republic of Bashkortostan, 452607, Russian Federation
²Stavropol State Pedagogical Institute, 427A, Lenina ave., Stavropol, 355029, Russian Federation

E-mail: academic-mvd@mail.ru

Abstract. The paper outlines ways in which the Maple computer algebra system can be used to master one of the branches of discrete mathematics – set theory. Computer algebra systems (CAS) expand the possibilities of creating, applying and using mathematical models on a daily basis by engineers, researchers, and foster the improved training of students in a wide range of disciplines. In addition, computer technology can help implement such methods as genetic approach and the use of various techniques to representing target objects in education. The most suitable for teaching discrete mathematics and mathematics at large are computer algebra systems such as Maple and Mathematica. The paper provides a Maple framework designed for the initial stages of mastering discrete mathematics within university curriculum, in particular, for working with the basic concepts and laws of set theory.

1. Introduction
Recently, various information technologies, including Maple, Mathematica, and other computer algebra systems, have been used in education. These tools are used in class to solve complex tasks, to perform large and complex mathematical transformations, etc. avoiding computational errors. Using such tools, the student feels more confident and spends less time solving a task set. Maple Computer Mathematics System (CM System) enables the creation of one’s own procedure library. In developing this, the student acquires the ability to program masterfully. Once created, the procedures are used to reinforce various mathematical knowledge and skills. Today, mathematics delivered at higher education institutions cannot be imagined without such a novel computer mathematics system as Maple [1]. This is one of the most powerful computer maths systems. It covers many branches of mathematics and can be used both in educational and serious scientific dimensions. Maple system enables to upgrade the methodology for practical and laboratory classes for students in a wide range of disciplines. It is quite cost-efficient to acquire skills in the Maple system, and the element of research activity has a very stimulating effect on students [5]. The Maple system gives students the opportunity to independently comprehend and monitor the basic theoretical principles of mathematical disciplines [6].

2. Materials and Methods
Discrete mathematics is a branch of mathematics that studies the properties of finite structures. Unlike discrete mathematics, classical mathematics generally considers infinite objects [7].
The course in discrete mathematics includes various topics embracing sets, relations, logic, Boolean algebra, number theory, graphs, and finite-state machines. But still, the starting point and basis for both discrete and continuous mathematics are sets.

Currently, the issues related to improving the efficiency and depth of mathematical training for students in the context of new information and communication technologies are becoming increasingly relevant. In this respect, one of the priorities is to increase the share of mathematical tasks and to use computer algebra (mathematics) systems (CAS) in teaching process. Many researchers believe that the most promising applications involve Mathematica and Maple computer algebra systems, the leaders among application software of this kind [2, 6]. Set theory is a branch of mathematics, which is studied not only in the framework of discrete mathematics [8], but is also widely used in IT disciplines [9, 10]. A wide applicability of set theory to various fields of knowledge can be due to the fact that various objects can act as elements of sets [3]. For example, for mathematics, such elements are most often algebraic expressions, functions, geometric shapes, numbers, and for computer science – information properties, information representation, programming languages, for biology – bacteria, fungi, plants, animals, for linguistics – content words and auxiliary parts of speech, alphabet, etc. The meta-subject matter of set theory implies studying the properties of sets without addressing the nature of their constituent elements.

Despite the relevance of set theory, reflected in the key training content for specialists in both mathematical and computer science directions, it is not yet properly delivered. The material below demonstrates the possibilities provided by the Maple computer algebra system for defining sets, proposing the laws and properties of sets, etc.

3. Numerical modeling
To demonstrate the solution procedure in set theory, the performance of operations on sets and the laws of set theory, a visual representation is used based on the diagrams where sets are presented in the form of circles. These diagrams are called Venn diagrams or Euler diagrams. However, there is another way to illustrate the execution of operations on sets and the laws of set theory, which, in our opinion, is more visual and allows a further more reasonable rationale for the organization and execution of logical operations. This is the so-called table of membership in sets, which represents all possible cases of a selected element appearing in sets $A$ and $B$. The results of membership relations are put in the first two columns of the table as per the following rule: 1 – if an element is a member of this set, 0 – if it is not. For the two sets there are four cases or four rows in the table. The columns corresponding to the operations denoted $A \cup B$, $A \cap B$, $A \setminus B$ are filled in, pursuant to the purpose of these operations.

Consider the proof of De Morgan’s law: $\overline{A \cup B} = \overline{A} \cap \overline{B}$.
It is possible to prove the validity of the law in various ways, one of which is through the use of the table of membership.

| Row No. | Set $A$ | Set $B$ | $\overline{A \cup B}$ | $\overline{A} \cap \overline{B}$ |
|---------|---------|---------|------------------------|------------------------|
| 1       | 1       | 1       | 0                      | 0                      |
| 2       | 1       | 0       | 0                      | 0                      |
| 3       | 0       | 1       | 0                      | 0                      |
| 4       | 0       | 0       | 1                      | 1                      |

The values for the last two columns of Table 1 can be determined as follows. A universal set is a set that contains a set $A$ and a set $B$, i.e. contains the rows labeled 1, 2, 3, 4, and sets $A$ and $B$ – rows with membership numbers “1”. Thus, the set $A = \{1, 2\}$, and the set $B = \{1, 3\}$, define them in the Maple computer algebra system:
with(Logic):
U := {1, 2, 3, 4}; # Specification of a universal set U for proving de Morgan’s law

The output value displayed on the screen: U := {1, 2, 3, 4}

set_A := {1, 2};
The output value displayed on the screen: set_A := {1, 2}

set_B := {1, 3};
The output value displayed on the screen: set_B := {1, 3}

To prove De Morgan’s law using the table of elements appearing in sets, it is necessary to determine the sets \( \overline{A} \cup \overline{B} \) and \( \overline{A} \cap \overline{B} \). The set \( \overline{A} \cup \overline{B} \) corresponds to a notation in the Maple computer algebra system:

\[
K_1 := U \min (\text{set}_A \cup \text{set}_B); \quad \# \text{Specification of set } U, \text{denoted } U \\setminus (A \cup B)
\]

The output value displayed on the screen: \( K_1 := \{4\} \)

A set, denoted \( \overline{A} \cap \overline{B} \), can be derived in Maple as follows:

\[
K_2 := (U \min \text{set}_A) \inter (U \min \text{set}_B); \quad \#
\]

Specification of a set, denoted \( (U \setminus A) \cap (U \setminus B) \)

The output value displayed on the screen: \( K_2 := \{4\} \)

The above can indicate that the operations performed are equivalent. Maple-enabled computations correspond to the output values presented in Table 2.

| Column No. | Set A | Set B | \( \overline{A} \cup \overline{B} \) | \( \overline{A} \cap \overline{B} \) |
|------------|-------|-------|----------------|----------------|
| 1          | 1     | 1     | 0              | 0              |
| 2          | 1     | 0     | 0              | 0              |
| 3          | 0     | 1     | 0              | 0              |
| 4          | 0     | 0     | 1              | 1              |

A check-up can also be done using the Maple-Logic corpus:

\[
\text{Equivalent}(K_1, K_2);
\]

The output value displayed on the screen: \textit{true}

The above computations clearly demonstrate the validity of De Morgan’s law and provide, through a similar algorithm, a framework for research (propositional calculus), and other laws of set theory. Students can verify the validity of theorems and laws of set theory not only through membership tables or other methods, but also through a computer algebra system they are familiar with (in our case, the Maple system).

De Morgan’s law can be written as:

\[
\text{De}_\text{-}M\text{\text{O}}\text{R}_{\text{G}}\text{AN} := \text{proc}(L)
\text{local } K, G;
\text{if op(0,L)=`minus` and nops(L)=2 }
\text{and op(1,L)=U }
\text{then}
K := op(2,L);
\text{if op(0,K)=`union`}
\text{then ((U minus op(1,K))intersect (U minus op(2,K)))}
\text{else L}
\end{proc}
\]
end if
end if
end proc:

The output value is:
> De_Morgan(U minus (A union B));
(U \setminus A) \cap (U \setminus B)

The distributive law can be written as:
> Distributive:=proc(L)
local n,p,q,r;
  n:=nops(L);
  if n=2 and nops(op(n,L))=2 then
    p:=op(1,L); q:=op(1,op(2,L)); r:=op(2,op(2,L));
    if op(0,L)=`intersect` and op(0,op(2,L))=`union` then
      p intersect q union (p intersect r)
    elif op(0,L)=`union` and op(0,op(2,L))=`intersect` then
      (p union q) intersect (p union r)
    else L
  end if
else L
end if
end proc:

> Distributive(A intersect (B union C));
(A \cap (B \cup C)) 
> Distributive(A union (B intersect C));
(A \cup B) \cap (A \cup C)

With the above and other similar procedures that feature binary operations on sets, including union, intersection, difference, etc., various set theory tasks are solved, or the way these tasks can be solved is demonstrated. For example, the simplification of several sets.

4. Conclusions
The studies [3] present modeling as a key tool for solving discrete mathematical tasks. Engineers should be able to represent design objects using mathematical or physical models, which makes it possible to more accurately predict or study the behavior of objects (structures) of these models. The proposed technology allows students to work with more complex and realistic models, since a large number of changes can be made to adjust a model if need be.

In addition, CAS (Maple) helps students not only deal with propositions and arguments flow, but also make statements and then verify their correctness. Being good at programming is a good way to solve various teaching and learning challenges, since students feel more confident applying the pre-developed software, rather than working with purely mathematical symbols that represent discrete structures. For example, working with an unfamiliar notation will be a simple confirmation that this is exactly what denotes this symbol, and this allows for an enriched understanding of the problem situation, and allows the learner to focus on grasping the gist of theoretical calculations, rather than acquiring the theory solely [4]. The computational capabilities provided by Maple can be used to formulate, prove or refute statements, and to solve various set theory tasks in a formal representation, and to model various logical puzzles. In the future, based on the acquired knowledge and skills, they will be able to model logical connectives, the conscious use of Boolean algebra, and subsequently model tasks of multi-valued logic. This will be the next step in the study of discrete structures.
References

[1] Klima R, Sigmon N 2012. Cryptology, Classical and Modern, with Maplets. (Taylor & Francis/CRC Press, Boca Raton, FL).
[2] Char B W 2003 Maple 9 Learning Guide (Waterloo: Maplesoft)
[3] Latifah M A, Kilicman A, Zainuddin H 2005 Analysis of Results for Test 1, Test 2 and final Exam MTH3004 For May 2003/2004. In: Proc. Int.Adv.Tech.Congress, Putrajaya, Malaysia.
[4] ACM/IEEE. “Computer science curriculum 2008: An interim revision of CS 2001”, in Report from the interim review task force, ACM and the IEEE CS, Dec. 2008.
[5] Khuzina L B, Mukhametshin V S, Tyncherov K T and Shaikhutdinova A F 2018 On the choice of the oscillators' installation site International Journal of Civil Engineering and Technology 9(9) 1952-1959
[6] Gainsburg J 2013 Learning to model in engineering Mathematical Thinking and Learning 15 4.
[7] Shaidullina R M, Amirov A F, Muhametshin V S, Tyncherov K T 2017 Designing economic socialization system in the educational process of technological university European Journal of Contemporary Education 6 (1) 149–158 DOI: 10.13187/ejced.2017.1.149
[8] Rosen K H 2012 Discrete mathematics and its applications, 7th ed. (McGraw-Hill).
[9] Belabas K, Friedman E 2015 Computing the residue of the Dedekind zeta function Math. Comp. 84(291) 357–369.
[10] Biasse J-F, Fieker C 2014 Subexponential class group and unit group computation in large degree number fields LMS J. Comput. Math 17 385–403.