How to teach Mathematics to students of CS

Branimir Šešelja
Department of Mathematics and Informatics
Faculty of Sciences, University of Novi Sad

DAAD WS 2013

Bansko, *August 27, 2013*

Abstract

Abstract

This presentation summarize some experiences in teaching basic mathematical subjects to students of mathematics and to students of informatics.

Abstract

This presentation summarize some experiences in teaching basic mathematical subjects to students of mathematics and to students of informatics.
These were students of the first year of bachelor studies, hence with the similar previous mathematical education.

Abstract

This presentation summarize some experiences in teaching basic mathematical subjects to students of mathematics and to students of informatics.

These were students of the first year of bachelor studies, hence with the similar previous mathematical education.

We briefly analyze:

Abstract

This presentation summarize some experiences in teaching basic mathematical subjects to students of mathematics and to students of informatics.
These were students of the first year of bachelor studies, hence with the similar previous mathematical education.
We briefly analyze:

- Different importance and different usage of same math topics in further studies of mathematics and informatics.

Abstract

This presentation summarize some experiences in teaching basic mathematical subjects to students of mathematics and to students of informatics.
These were students of the first year of bachelor studies, hence with the similar previous mathematical education.
We briefly analyze:

- Different importance and different usage of same math topics in further studies of mathematics and informatics.
- Motivation of students of informatics, their questions and remarks. Examples.

Abstract

This presentation summarize some experiences in teaching basic mathematical subjects to students of mathematics and to students of informatics.
These were students of the first year of bachelor studies, hence with the similar previous mathematical education.
We briefly analyze:

- Different importance and different usage of same math topics in further studies of mathematics and informatics.
- Motivation of students of informatics, their questions and remarks. Examples.
- How to teach, which topics to present in details, what to avoid. Examples.

Abstract

This presentation summarize some experiences in teaching basic mathematical subjects to students of mathematics and to students of informatics.
These were students of the first year of bachelor studies, hence with the similar previous mathematical education.
We briefly analyze:

- Different importance and different usage of same math topics in further studies of mathematics and informatics.
- Motivation of students of informatics, their questions and remarks. Examples.
- How to teach, which topics to present in details, what to avoid. Examples.

  We give some general remarks.

Theoretical Foundations of Informatics 1

Theoretical Foundations of Informatics 1

- Logic

Theoretical Foundations of Informatics 1

- Logic
- Sets

Theoretical Foundations of Informatics 1

- Logic
- Sets
- Relations

Theoretical Foundations of Informatics 1

- Logic
- Sets
- Relations
- Functions

Theoretical Foundations of Informatics 1

- Logic
- Sets
- Relations
- Functions

Algebra for Informaticians

Theoretical Foundations of Informatics 1

- Logic
- Sets
- Relations
- Functions

Algebra for Informaticians

- Algebraic structures

Theoretical Foundations of Informatics 1

- Logic
- Sets
- Relations
- Functions

Algebra for Informaticians

- Algebraic structures
- Numbers

Theoretical Foundations of Informatics 1

- Logic
- Sets
- Relations
- Functions

Algebra for Informaticians

- Algebraic structures
- Numbers
- Polynomials

Theoretical Foundations of Informatics 1

- Logic
- Sets
- Relations
- Functions

Algebra for Informaticians

- Algebraic structures
- Numbers
- Polynomials
- Systems of linear equations; determinants, matrices

Formal aspect of teaching

Formal aspect of teaching

Does the structure of the institution have an impact to teaching?

Formal aspect of teaching

Does the structure of the institution have an impact to teaching?

- *Department of Mathematics and Informatics* or

Formal aspect of teaching

Does the structure of the institution have an impact to teaching?

- *Department of Mathematics and Informatics* or
- *Department of Informatics* and
  Department of Mathematics.

Formal aspect of teaching

Does the structure of the institution have an impact to teaching?

- *Department of Mathematics and Informatics* or
- *Department of Informatics* and
  Department of Mathematics.

  In the latter case:

Formal aspect of teaching

Does the structure of the institution have an impact to teaching?

- *Department of Mathematics and Informatics* or
- *Department of Informatics* and *Department of Mathematics*.

  In the latter case:
- Informatics Department having its own teachers in Math

Formal aspect of teaching

Does the structure of the institution have an impact to teaching?

- *Department of Mathematics and Informatics* or
- *Department of Informatics* and
  Department of Mathematics.

  In the latter case:
- Informatics Department having its own teachers in Math

  or

Formal aspect of teaching

Does the structure of the institution have an impact to teaching?

- *Department of Mathematics and Informatics* or
- *Department of Informatics* and *Department of Mathematics*.

  In the latter case:
- Informatics Department having its own teachers in Math

  or
- Math courses thought by teachers from the Math Department.

Some similarities and some differences

Some similarities and some differences

- Both, Math and CS students should obtain a certain level of *mathematical sophistication*, i.e., the ability to formalize concepts, work from definitions, think rigorously, reason concisely, and construct some theory.

Some similarities and some differences

- Both, Math and CS students should obtain a certain level of *mathematical sophistication*, i.e., the ability to formalize concepts, work from definitions, think rigorously, reason concisely, and construct some theory.
- Math students are supposed to deepen and develop the basic knowledge through advanced courses, therefore they will remember the basic topics and get familiar with them.

Some similarities and some differences

- Both, Math and CS students should obtain a certain level of *mathematical sophistication*, i.e., the ability to formalize concepts, work from definitions, think rigorously, reason concisely, and construct some theory.
- Math students are supposed to deepen and develop the basic knowledge through advanced courses, therefore they will remember the basic topics and get familiar with them.
- On the other hand, CS students are supposed to apply and use these topics; they may forget some (or many) definitions, theorems and proofs from these basic courses.

Some similarities and some differences

- Both, Math and CS students should obtain a certain level of *mathematical sophistication*, i.e., the ability to formalize concepts, work from definitions, think rigorously, reason concisely, and construct some theory.
- Math students are supposed to deepen and develop the basic knowledge through advanced courses, therefore they will remember the basic topics and get familiar with them.
- On the other hand, CS students are supposed to apply and use these topics; they may forget some (or many) definitions, theorems and proofs from these basic courses.

  However, what does matter is the *rigor*, *strictness* they learned that is supposed to help their careers and to last a lifetime.

First course (Logic, sets, relations, functions)

First course (Logic, sets, relations, functions)

Almost the same syllabi for both, students of Math and students of CS.

First course (Logic, sets, relations, functions)

Almost the same syllabi for both, students of Math and students of CS.

However, *logic* for mathematicians is a *logical introduction to mathematics*. They learn in details semantics of propositional and predicate logic. Their program does not contain axiomatic approach (propositional and predicate calculi).

First course (Logic, sets, relations, functions)

Almost the same syllabi for both, students of Math and students of CS.
However, *logic* for mathematicians is a *logical introduction to mathematics*. They learn in details semantics of propositional and predicate logic. Their program does not contain axiomatic approach (propositional and predicate calculi).
The reason is that they have *Mathematical logic* as a subject at the master level.

First course (Logic, sets, relations, functions)

Almost the same syllabi for both, students of Math and students of CS.
However, *logic* for mathematicians is a *logical introduction to mathematics*. They learn in details semantics of propositional and predicate logic. Their program does not contain axiomatic approach (propositional and predicate calculi).
The reason is that they have *Mathematical logic* as a subject at the master level.

Students of CS have logic only as a part of this first year course, no more logic after.

First course (Logic, sets, relations, functions)

Almost the same syllabi for both, students of Math and students of CS.
However, *logic* for mathematicians is a *logical introduction to mathematics*. They learn in details semantics of propositional and predicate logic. Their program does not contain axiomatic approach (propositional and predicate calculi).
The reason is that they have *Mathematical logic* as a subject at the master level.

Students of CS have logic only as a part of this first year course, no more logic after.
Therefore, they learn both, semantics and syntax; hence they have *propositional and predicate calculi* as formal theories.

"Mathematicians are proving theorems"

"Mathematicians are proving theorems"

What is *a proof*?

"Mathematicians are proving theorems"

What is *a proof*?

It is a method of establishing truth.

"Mathematicians are proving theorems"

What is *a proof*?

It is a method of establishing truth.
Though, it can be done in numerous ways in different branches.

"Mathematicians are proving theorems"

What is *a proof*?

It is a method of establishing truth.
Though, it can be done in numerous ways in different branches.

- Confirming a hypothesis by experiments.

"Mathematicians are proving theorems"

What is *a proof*?

It is a method of establishing truth.
Though, it can be done in numerous ways in different branches.

- Confirming a hypothesis by experiments.
- By jury trial.

"Mathematicians are proving theorems"

What is *a proof*?

It is a method of establishing truth.
Though, it can be done in numerous ways in different branches.

- Confirming a hypothesis by experiments.
- By jury trial.
- By statistical analysis of many samples of evidence.

"Mathematicians are proving theorems"

What is *a proof*?

It is a method of establishing truth.
Though, it can be done in numerous ways in different branches.

- Confirming a hypothesis by experiments.
- By jury trial.
- By statistical analysis of many samples of evidence.
- By self conviction: "*I know* that this is true."

"Mathematicians are proving theorems"

What is *a proof*?

It is a method of establishing truth.
Though, it can be done in numerous ways in different branches.

- Confirming a hypothesis by experiments.
- By jury trial.
- By statistical analysis of many samples of evidence.
- By self conviction: "*I know* that this is true."
- By the Word of God...

"Mathematicians are proving theorems"

What is *a proof*?

It is a method of establishing truth.
Though, it can be done in numerous ways in different branches.

- Confirming a hypothesis by experiments.
- By jury trial.
- By statistical analysis of many samples of evidence.
- By self conviction: "*I know* that this is true."
- By the Word of God...
- ...

In mathematics, a proof is a verification of a proposition by a finite chain of logical deductions from a basic set of axioms.

In mathematics, a proof is a verification of a proposition by a finite chain of logical deductions from a basic set of axioms.

In their basic math courses, CS students learn *definitions, axioms, theorems and proofs*.

In mathematics, a proof is a verification of a proposition by a finite chain of logical deductions from a basic set of axioms.

In their basic math courses, CS students learn *definitions, axioms, theorems and proofs*.
Should they?

In mathematics, a proof is a verification of a proposition by a finite chain of logical deductions from a basic set of axioms.

In their basic math courses, CS students learn *definitions, axioms, theorems and proofs*.
Should they?

The link to their studies is that e.g., in programming, the analogs of these are data structures, assumptions, procedures, algorithms ...

In mathematics, a proof is a verification of a proposition by a finite chain of logical deductions from a basic set of axioms.

In their basic math courses, CS students learn *definitions, axioms, theorems and proofs*.
Should they?

The link to their studies is that e.g., in programming, the analogs of these are data structures, assumptions, procedures, algorithms ...

If a student is confused about the difference between *necessary and sufficient conditions* he/she is likely to misunderstand an *if ... then* statement in a program.

In mathematics, a proof is a verification of a proposition by a finite chain of logical deductions from a basic set of axioms.

In their basic math courses, CS students learn *definitions, axioms, theorems and proofs*.
Should they?

The link to their studies is that e.g., in programming, the analogs of these are data structures, assumptions, procedures, algorithms ...

If a student is confused about the difference between *necessary and sufficient conditions* he/she is likely to misunderstand an *if ... then* statement in a program.
A student who is not able to write a one-page proof of a theorem, could hardly pass trough, analyze and debug huge programs, having millions of lines.

Teaching Mathematics to students of IT and CS

Teaching Mathematics to students of IT and CS

Some experiences

Teaching Mathematics to students of IT and CS

Some experiences

The students have necessary skills in order to follow the lectures.

Teaching Mathematics to students of IT and CS

Some experiences

The students have necessary skills in order to follow the lectures.

They know why they study the corresponding program.

Teaching Mathematics to students of IT and CS

Some experiences

The students have necessary skills in order to follow the lectures.

They know why they study the corresponding program.

However, they are not motivated to learn mathematics (in particular IT students).

Teaching Mathematics to students of IT and CS

Some experiences

- The students have necessary skills in order to follow the lectures.
- They know why they study the corresponding program.
- However, they are not motivated to learn mathematics (in particular IT students).
- Therefore, the most important task at the beginning is to make them believe that mathematical topics are necessary.

Teaching Mathematics to students of IT and CS

Some experiences

The students have necessary skills in order to follow the lectures.

They know why they study the corresponding program.

However, they are not motivated to learn mathematics (in particular IT students).

Therefore, the most important task at the beginning is to make them believe that mathematical topics are necessary.

How?

The easiest way is by brute force: "*Math topics do belong to your study program*".

The easiest way is by brute force: "*Math topics do belong to your study program*".

Of course, there are more convincing arguments.

The easiest way is by brute force: "*Math topics do belong to your study program*".
Of course, there are more convincing arguments.

Examples

The easiest way is by brute force: "*Math topics do belong to your study program*".

Of course, there are more convincing arguments.

Examples

Induction

The easiest way is by brute force: "*Math topics do belong to your study program*".
Of course, there are more convincing arguments.

Examples

Induction

- Correct example: *Dominos*

The easiest way is by brute force: "*Math topics do belong to your study program*".

Of course, there are more convincing arguments.

Examples

Induction

- Correct example: *Dominos*
- False: *All persons in a group of n people have the same hair color.*

The easiest way is by brute force: "*Math topics do belong to your study program*".
Of course, there are more convincing arguments.

Examples

Induction

- Correct example: *Dominos*
- False: *All persons in a group of n people have the same hair color.*
- False: *I can carry all sand at a beach.*

The easiest way is by brute force: "*Math topics do belong to your study program*".
Of course, there are more convincing arguments.

Examples

Induction

- Correct example: *Dominos*
- False: *All persons in a group of n people have the same hair color.*
- False: *I can carry all sand at a beach.*

Proof by Reductio ad Absurdum:

The easiest way is by brute force: "*Math topics do belong to your study program*".
Of course, there are more convincing arguments.

Examples

Induction

- Correct example: *Dominos*
- False: *All persons in a group of n people have the same hair color.*
- False: *I can carry all sand at a beach.*

Proof by Reductio ad Absurdum:

- Equation $x^2 = 2$ is not solvable over rational numbers.

The easiest way is by brute force: "*Math topics do belong to your study program*".
Of course, there are more convincing arguments.

Examples

Induction

- Correct example: *Dominos*
- False: *All persons in a group of n people have the same hair color.*
- False: *I can carry all sand at a beach.*

Proof by Reductio ad Absurdum:

- Equation $x^2 = 2$ is not solvable over rational numbers.

Deduction (algorithm)

The easiest way is by brute force: "*Math topics do belong to your study program*".
Of course, there are more convincing arguments.

Examples

Induction

- Correct example: *Dominos*
- False: *All persons in a group of n people have the same hair color.*
- False: *I can carry all sand at a beach.*

Proof by Reductio ad Absurdum:

- Equation $x^2 = 2$ is not solvable over rational numbers.

Deduction (algorithm)

- Switch problem.

Sets, relations and functions

Sets, relations and functions

These topics are usually well accepted by students.

Sets, relations and functions

These topics are usually well accepted by students.

E.g., presenting relations by ordered pairs, triples, ... , *n*-tuples, can be connected with databases.

Sets, relations and functions

These topics are usually well accepted by students.

E.g., presenting relations by ordered pairs, triples, ... , n-tuples, can be connected with databases.
Equivalence relations and partitions are associated to e.g., pattern recognition.

Sets, relations and functions

These topics are usually well accepted by students.

E.g., presenting relations by ordered pairs, triples, ... , *n*-tuples, can be connected with databases.
Equivalence relations and partitions are associated to e.g., pattern recognition.
Indexed families of sets $\longrightarrow$

Sets, relations and functions

These topics are usually well accepted by students.

E.g., presenting relations by ordered pairs, triples, ... , n-tuples, can be connected with databases.
Equivalence relations and partitions are associated to e.g., pattern recognition.
Indexed families of sets $\longrightarrow$ direct product of a family $\longrightarrow$

Sets, relations and functions

These topics are usually well accepted by students.

E.g., presenting relations by ordered pairs, triples, ... , n-tuples, can be connected with databases.
Equivalence relations and partitions are associated to e.g., pattern recognition.
Indexed families of sets $\longrightarrow$ direct product of a family $\longrightarrow$ (multi dimensional) sequences $\longrightarrow$

Sets, relations and functions

These topics are usually well accepted by students.

E.g., presenting relations by ordered pairs, triples, ... , n-tuples, can be connected with databases.
Equivalence relations and partitions are associated to e.g., pattern recognition.
Indexed families of sets $\longrightarrow$ direct product of a family $\longrightarrow$ (multi dimensional) sequences $\longrightarrow$ storing data, tables, matrices etc.

Second course (Structures, numbers, polynomials, linear algebra)

Second course (Structures, numbers, polynomials, linear algebra)

Algebraic structures (groupoid, semigroup, group, ring, field, vector space) are connected and illustrated by numbers.

Second course (Structures, numbers, polynomials, linear algebra)

Algebraic structures (groupoid, semigroup, group, ring, field, vector space) are connected and illustrated by numbers.
In addition, finite groups are connected to Latin squares, and to decoding of linear codes by use of Lagrange theorem.

Second course (Structures, numbers, polynomials, linear algebra)

Algebraic structures (groupoid, semigroup, group, ring, field, vector space) are connected and illustrated by numbers.
In addition, finite groups are connected to Latin squares, and to decoding of linear codes by use of Lagrange theorem.
In Number theory, there are many possibilities to find algorithms (divisibility, Euclidean algorithm, ...)

Second course (Structures, numbers, polynomials, linear algebra)

Algebraic structures (groupoid, semigroup, group, ring, field, vector space) are connected and illustrated by numbers.
In addition, finite groups are connected to Latin squares, and to decoding of linear codes by use of Lagrange theorem.
In Number theory, there are many possibilities to find algorithms (divisibility, Euclidean algorithm, ...)
Polynomials are presented as terms and as sequences over a field (ring).

Some questions by students

Some questions by students

- *Do we really need to prove this?*

Some questions by students

- *Do we really need to prove this?*
- *What is essentially a complex number (in particular the number* $\mathbf{i}$*)?*

Some questions by students

- *Do we really need to prove this?*
- *What is essentially a complex number (in particular the number* $\mathbf{i}$*)?*
- *If practically never we get an irrational number among some data, then why do we need these numbers?*

Some questions by students

- *Do we really need to prove this?*
- *What is essentially a complex number (in particular the number* **i***)?*
- *If practically never we get an irrational number among some data, then why do we need these numbers?*
- *So you have proved that this is true, but is there also a proof that it is not?*

Some questions by students

- *Do we really need to prove this?*
- *What is essentially a complex number (in particular the number* $\mathbf{i}$*)?*
- *If practically never we get an irrational number among some data, then why do we need these numbers?*
- *So you have proved that this is true, but is there also a proof that it is not?*
- *How could* $0.\overline{99}$ *be equal* 1*, since it is obviously less.*

According to experience: How to teach Mathematics for students of CS?

According to experience: How to teach Mathematics for students of CS?

- At the beginning: try to convince them that basic mathematics is necessary:

According to experience: How to teach Mathematics for students of CS?

- At the beginning: try to convince them that basic mathematics is necessary:

  By presenting attractive, simple examples connected to CS, having mathematical explanation.

According to experience: How to teach Mathematics for students of CS?

- At the beginning: try to convince them that basic mathematics is necessary:

  By presenting attractive, simple examples connected to CS, having mathematical explanation.

  By recalling knowledge from secondary school (through examples).

According to experience: How to teach Mathematics for students of CS?

- At the beginning: try to convince them that basic mathematics is necessary:

  By presenting attractive, simple examples connected to CS, having mathematical explanation.

  By recalling knowledge from secondary school (through examples).

  By claiming that this will not be difficult.

When presenting a new topic

When presenting a new topic

- Start with an example in which the new topic appears (implicitly).

When presenting a new topic

- Start with an example in which the new topic appears (implicitly).
- Then describe the new topic colloquially, without (too many) formulas.

When presenting a new topic

- Start with an example in which the new topic appears (implicitly).
- Then describe the new topic colloquially, without (too many) formulas.
- Again an example.

When presenting a new topic

- Start with an example in which the new topic appears (implicitly).
- Then describe the new topic colloquially, without (too many) formulas.
- Again an example.
- Finally give the **precise, correct mathematical definition, formulation of the theorem**...

When presenting a new topic

- Start with an example in which the new topic appears (implicitly).
- Then describe the new topic colloquially, without (too many) formulas.
- Again an example.
- Finally give the **precise, correct mathematical definition, formulation of the theorem**...
- If the theorem has to be proved, first illustrate the proof throughout an example.

When presenting a new topic

- Start with an example in which the new topic appears (implicitly).
- Then describe the new topic colloquially, without (too many) formulas.
- Again an example.
- Finally give the **precise, correct mathematical definition, formulation of the theorem**...
- If the theorem has to be proved, first illustrate the proof throughout an example.
- Then present the **correct proof**.

When presenting a new topic

- Start with an example in which the new topic appears (implicitly).
- Then describe the new topic colloquially, without (too many) formulas.
- Again an example.
- Finally give the **precise, correct mathematical definition, formulation of the theorem**...
- If the theorem has to be proved, first illustrate the proof throughout an example.
- Then present the **correct proof**.
- Then again the above example.

When presenting a new topic

- Start with an example in which the new topic appears (implicitly).
- Then describe the new topic colloquially, without (too many) formulas.
- Again an example.
- Finally give the **precise, correct mathematical definition, formulation of the theorem**...
- If the theorem has to be proved, first illustrate the proof throughout an example.
- Then present the **correct proof**.
- Then again the above example.
- Avoid philosophical discussions, relay to practical explanations.

- For exercises, try to find problems in which students should discover a necessity of using some mathematical tool.

- For exercises, try to find problems in which students should discover a necessity of using some mathematical tool.
- Present problems with several possible approaches and encourage students to explore and find an appropriate algorithm.

- For exercises, try to find problems in which students should discover a necessity of using some mathematical tool.
- Present problems with several possible approaches and encourage students to explore and find an appropriate algorithm.
- *A suggestion by a colleague:* try to write on a board, even if you use a beamer presentation.

- For exercises, try to find problems in which students should discover a necessity of using some mathematical tool.
- Present problems with several possible approaches and encourage students to explore and find an appropriate algorithm.
- *A suggestion by a colleague:* try to write on a board, even if you use a beamer presentation.

  In this way, students will believe you.

- For exercises, try to find problems in which students should discover a necessity of using some mathematical tool.
- Present problems with several possible approaches and encourage students to explore and find an appropriate algorithm.
- *A suggestion by a colleague:* try to write on a board, even if you use a beamer presentation.

  In this way, students will believe you.

  To them, it would mean that they could do the same, i.e., that they can learn the topic.

- For exercises, try to find problems in which students should discover a necessity of using some mathematical tool.
- Present problems with several possible approaches and encourage students to explore and find an appropriate algorithm.
- *A suggestion by a colleague:* try to write on a board, even if you use a beamer presentation.

  In this way, students will believe you.

  To them, it would mean that they could do the same, i.e., that they can learn the topic.
- Encourage students to **understand** topics, formulations, theorems...

- For exercises, try to find problems in which students should discover a necessity of using some mathematical tool.
- Present problems with several possible approaches and encourage students to explore and find an appropriate algorithm.
- *A suggestion by a colleague:* try to write on a board, even if you use a beamer presentation.

  In this way, students will believe you.

  To them, it would mean that they could do the same, i.e., that they can learn the topic.
- Encourage students to **understand** topics, formulations, theorems...
- At the oral exam, do not insist (to much) on formal proofs, require understanding.

- For exercises, try to find problems in which students should discover a necessity of using some mathematical tool.
- Present problems with several possible approaches and encourage students to explore and find an appropriate algorithm.
- *A suggestion by a colleague:* try to write on a board, even if you use a beamer presentation.

  In this way, students will believe you.

  To them, it would mean that they could do the same, i.e., that they can learn the topic.
- Encourage students to **understand** topics, formulations, theorems...
- At the oral exam, do not insist (to much) on formal proofs, require understanding.
- Have permanent contact with colleagues teaching CS subjects.

Thank you, that was all!