Practical information course Theoretical Computer Science 2017-2018

Bruno Bauwens

January 21, 2018

This[1] course is given to first year Phd students computer science of the Higher School of Economics in Moscow. The instructor is Bruno Bauwens, and you can reach me on BBauwens gmail. com. Lectures are in Kochnovskiy Proezd 3, room 303, on Mondays from 18h10-21h00. All information about the course can be found on my webpage at HSE: `https:\\www.hse.ru\en\org\persons\160550073#teaching` .

The goals of the course are to

- learn basic concepts from the theory of computation,
- train abstract thinking and problem solving,
- train writing, explaining and presenting abstract ideas (in English).

Grades. Students can earn in total at most 40 points.

16 points for the project (paper + lecture, see further),
12 points for the two intermediate exams, and
12 points for the final exam.

The official grade is on a scale of 10. For scores strictly below 20, your score will be divided by 4 and rounded down. Otherwise, it will be divided by 4 and rounded up. To pass for the course you need an official grade of at least 4/10 (which corresponds to $\geq 16/40$). The rules to keep your grant are unclear. However, you need at least 6/10 (which corresponds to $< 20/40$), to keep your stipend slightly higher. The rules sometimes change, contact Timofei Nikulichev for the most recent information about this.

People who regularly participate in discussions during the class, or regularly send me mistakes in the lecture notes receive a bonus point.

Books. Most of the course is based on the Michael Sipser. Introduction to the Theory of Computation. Cengage Learning, 2012.

Exams. During the exam you can bring handwritten notes and printed lecture materials. You will also be given a computer with all lecture notes and Sipser's book. The exam will contain

[1] First version: 15th of Januari. Update on the 21st of Januari : marks and grant, extra exams, and list of conferences for the project.

exercises that are similar to the ones solved in class and similar to the problems at the end of each chapter in Sipser's book.

The intermediate exam covers the contents of the 6 first lectures: regular languages, computability and the complexity classes P and NP. The final exam covers the advanced topics.

There will be an additional exam, probably in may for those who do longterm visits in April. Finally, there will be an additional exam in August for those who did not pass in the other exam.

The project

Students are expected to write and give a lecture about a *mathematical* result related to any area of computer science, discrete mathematics, applied mathematics, engineering or statistics. The goal is to transform a result from a paper written for specialists to a paper that is readable for the other students in the class. Thus, the goal is to write something that is between lecture notes and a research paper.

The topic. The paper should be mathematical: it should contain definitions, one or two theorems, and proofs. The most important part for the grade is given for the proof. The topic should either be a paper from one of the main conferences in theoretical computer science, some topic related to the course or some other topic related to your research. The project can not be anything that is typical undergraduate material (in a high quality university). Here is a list of conferences you might use to choose a paper.

- FOCS: (Symposium on the) Foundations of Computer Science,
- STOCS: Symposium on the Theory of Computing,
- STACS: Symposium on Theoretical Aspects of Computer Science,
- ICALP: International colloquium on Automata, Languages and Programming,
- CCC: Conference on Computational Complexity,
- COLT: Computational Learning Theory
- CSR: Computer science symposium in Russia

FOCS and STOCS are the most prestigious conferences in theoretical computer science. Students with a weaker background in mathematics might prefer to present an algorithm, for example from the STACS conference.

You are allowed to deviate from the guidelines above, for example you can discuss a topic from the book "Computational Complexity a modern approach" by S. Arora and B. Barak. Or you can discuss a heuristic deviation of a machinelearning algorithm that at some point uses a smaller theorem. ICML and NIPS are very nice machinelearning conferences. In such cases, you should discuss or email your proposal for a project to the teacher. Note that the paper should be theoretical. You can discuss practical results to make the paper more interesting, but the majority of the marks are given for the exposition of a proof.

To download papers good websites are scholar.google.com and sci-hub.cc. For books, you can use gen.lib.rus.ec, libgen.io and libgen.pw.

Structure of the paper.

- First write clearly what is the main source of information from which you obtained the presented materials.
- A short introduction to explain general goals of the subfield of research, and some motivation for the result that is presented.
- The definitions needed to understand the main theorem(s), and these theorems. The paper should be readable by a typical beginning graduate student.
- The proof.

For easier grading, present definitions at the place you need them (don't start with a big paragraph full of notations that are used somewhere in the paper). Also, add line numbers to your paper. For Latex users add: "\usepackage{lineno}" and "\linenumbers" in the beginning of the tex-file.

Criteria for grading the paper. The paper should be at least 5 pages. The grade does not depend on the length of the paper. Criteria for the grade:

- Satisfies the conditions of the project (there is a clear theorem, enough background information for general audience, at least 5 pages, no undergrad materials, etc.)
- Clarity of the paper.
- Difficulty of the proof of the main theorem(s).
- Does the student have a good understanding of the paper, this is also checked during the presentation through questions.
- How much the paper changed from the original version:
  did the author add: background information for a general audience, **clarifying examples**, high level explications, detailed mathematical explications,
- Quality of English.

It is highly recommended to add examples wherever possible. One should write in formal English. For example do not use abbreviations: the words "don't" and "let's" should be written in full: "do not" and "let us". Pay attention to the choice of the articles "the" and "a". Never copy sentences from any source, unless the copied part is clearly marked and the source acknowledged. You are allowed to copy definitions, lemmas and theorems from the main reference. You can not copy anything from the proofs.

After the paper is submitted, the teacher will write recommendations on how to improve the paper (parts that are unclear, errors against English grammar, etc.). Then the student can submit a final version. Both submissions of the paper counts towards the grade, so to get through the course it is better to submit an optimized paper.

The lecture practal. The lecture should take about 50 minutes. Students will be divided in groups of around 6-7 students who present their work during one day, typically a Saturday. Presence at all lectures of your group is mandatory. Only the blackboard should be used to explain the proof. For other parts of the lecture, the student may use slides.

Criteria for grading the lecture. The goal of the lecture is to give a general overview of your paper and to explain a part of your paper in a clear and interactive way. Your talk

should cover definitions and some proofs or a part of the proof. The mark will be given after answering the following questions.

- Has the speaker profound understanding of the paper?
- Does he ask questions to check whether the audience understand what he explained?
- Are there examples to illustrate definitions (and proofs)?
- Are there nice schematic overviews? Are there good informal descriptions?
- Is there a formal presentation of the proofs that is as clean as possible?
- Does the speaker succeed to have many discussions with the audience?
- Does the speaker explain non-trivial mathematical materials?

The whiteboard should be used to present the proof. If needed, slides can be used to present motivation, background and definitions. The speaker is supposed to know all materials and can not hold sheets during the presentation. (Except for very specific purposes, for example if some big table is needed, or there is an example with a cumbersome formula.) Recall that other students might have a different background. Decrease the speed if you might loose the audience, it is not important to cover all of the paper, especially for the more complex papers.

Grades. The 16 points for the project are distributed as follows:

- 4 points for the initial version of the paper,
- 4 points for the final version of the paper,
- 6 points for the presentation,
- 2 points for participating in discussions during the presentations in your group.

Tip: Latex is an excellent editor to type mathematical texts.

Enjoy studying and doing your project,

Bruno Bauwens