Why Study Automata?

**What you had so far in most CS classes.** In most Computer Science classes:

- you are given a problem, and
- you need to design a program to solve this problem.

In all such cases:

- you are sure that the problem has a solution, it is just a question of designing an appropriate program, and
- once you have a program that works, you are happy, your instructor is happy, your TA is happy – and how exactly this program is implemented in the actual computer, is rather irrelevant.

**Not all problems are solvable – and the related first objective of the class.** Problems that are assigned in classes are definitely solvable. However, when you graduate and start working, problems will come from practical needs, not from an instructor. And in real life, not all problems are solvable by an algorithm.

This may be because the problem is formulated in too general a form. So, from the practical viewpoint, it is important:

- to understand that the general problem is not solvable – so as not to waste time trying to find a solution – and
- to come up with a more specific problem that *does* have a solution.

This may also be because the problem is asking for too much: e.g., often:

- finding an *optimal* solution is not possible – since this would require an unrealistic amount of computation time,
- but it may be possible to come up with a good enough solution.

So, *the first objective of the Automata class is:*

- to teach the students that some problems are not solvable – and why, and
- to help students develop a general understanding of which general problems are solvable and which are not.
Caution. Of course, in a one-semester class, we will only learn the basics. To get a better understanding of what is computable and what is not, you can take a graduate class on Theory of Computation – or you can take an appropriate special topics class when such a class is offered.

What do we need to study to understand which problems are solvable and which are not. Deciding which problems are algorithmically solvable and which are not is important. But one cannot decide on this by just looking at possible programs – we need to get a really good understanding of what exactly a computer does when you run a program written in a programming language. This is a complex process, and understanding this process is what we will do most of the semester.

So, we arrive at the second objective of this class.

How programs are compiled – the second objective of this class. To an average person, a computer is a magic box:

- you write a program, and
- somehow a computer understands what needs to be done.

Yes:

- sometimes the computer is dumb: you miss a semi-colon, and it produces God knows what,
- but when it does work correctly, it is magic.

How does the computer go from a program to a specific sequence of actions? In other words, how does it compile the program? To us, it may be magic, but there is nothing magical about it: there are straightforward algorithms.

So, the second objective of the class is to understand how program are compiled.

Caution. In a one-semester class, we will only study the main ideas behind compiling. Of course, real compilers use many other ideas:

- the basic ideas that we study in this class will usually be enough to design a simple compiler,
- but to design a practically efficient compiler, many more ideas are needed.

If you are really interested in practical compiling, you may want to take a Compiler class whenever we offer it.

Is this easy? No, this class will not be easy. For most Computer Science students, this is the only theoretical class, the only class when we will have:

- not only to come up with algorithms – this is we are all good at,
- but also come up with proofs that some problems do not have a solution – and this will be, to many students, a new and challenging experience.
How to be successful in this class. The usual advice is:

- to submit homeworks on time if possible; it is usually more difficult to catch up;

- to ask questions if something is not clear; students are sometimes reluctant to ask questions, thinking that they are the only ones in the class who do not understand – this is not true: if you do not understand, probably half of the class does not understand either, so your question helps others; good questions will give you extra points; even without these extra points, questions are usually correlated with the grade in class: the more question the student asks, the better the overall grade;

- to form study groups – this is what most faculty did when we were students; studying together, in person or remotely, helps: usually, for most topics, at least one person in a group has a better understanding of this topic and can help others.

How to study?

- reading material is important, but

- the only way to learn is to do exercises; do the homeworks, do class exercises, there are many similar problems in the textbook; there are many solutions in website of classes from previous semesters; do not hesitate to ask for help if needed.

Before a test, do not just read the material again and again: try to do sample problems, just to do problems from the previous semesters’ tests, ask if something is not clear. An instructor would rather spend time answering questions before the test than explain after the test what was wrong.

Some advice is specific for this class:

- you will be given some proofs, and you will be asked to prove similar results; the important thing is to understand every single step in the given proof; if one step is missing or not clear, we do not have a proof; do not hesitate to ask questions if something is not clear, we all were in this situation when studying proofs;

- do not improvise: in English classes, you are taught (correctly) that you cannot just reproduce the words from the textbook in your essays, you have to reformulate the ideas in your own words; with definitions and proofs, this is not true: you have to use the exact language, otherwise, your reformulation may be wrong.

Can all the students succeed? Absolutely yes, year after year the vast majority of the students pass this class. You all passed very complex classes before, both in Math and in CS, this shows that your brains can solve complex problems.

The only way to fail is not to turn in homeworks, not to study.

*We expect all of you to succeed!*