Theory of Computation

Module Description

Module code	CS203-S28	**Credits**	7 ECTS
Programme	Foundation programme	**Level**	Semester 4
Semester	Spring 2028	**Prerequisites**	MAT101-F26 Discrete Structures; CS232-F27 Data Structures and Algorithms II
Module leader	Dr. Ali Khudiyev	**Email**	ali.khudiyev@wosm.academy
Office hours	Monday–Friday, 10:00–17:00 (Baku time, UTC+4)		

1. Module Overview

Theory of Computation asks what computers can do in principle and what they can do efficiently. It studies abstract models of computation, from finite automata to Turing machines, and the classes of languages they recognise. Finite automata, met briefly in Digital Design and Computer Architecture I, are treated here in full.

The module has three parts. *Automata and formal languages* covers regular and context-free languages, which underlie pattern matching, lexing and parsing. *Computability* shows that some problems, such as the halting problem, cannot be solved by any program. *Complexity* makes precise the distinction between tractable and intractable problems introduced in Data Structures and Algorithms II.

2. Aims

- Introduce the standard mathematical models of computation and the relationships between them.
- Show the limits of computation through rigorous proofs of undecidability.
- Give a precise foundation for complexity classes and NP-completeness.
- Develop your ability to write careful proofs about formal objects.

3. Learning Outcomes

After completing this course, you will be able to:

LO1. Construct deterministic and nondeterministic finite automata and regular expressions, convert between them, and minimise automata.

LO2. Prove that languages are not regular using the pumping lemma and the Myhill–Nerode theorem.

LO3. Construct context-free grammars and pushdown automata, transform grammars into Chomsky normal form, parse with the CYK algorithm, and prove that languages are not context-free.

LO4. Describe Turing machines and equivalent models, explain the Church–Turing thesis, and prove problems undecidable by diagonalisation and reduction.

LO5. Place problems in complexity classes such as P, NP and PSPACE, and prove that problems are NP-complete by reduction.

4. Syllabus

Unit	Topic
1	Alphabets, words and languages; proof techniques revisited
2	Deterministic finite automata
3	Nondeterminism and the subset construction
4	Regular expressions and Kleene's theorem
5	Non-regular languages: pumping lemma, Myhill–Nerode, minimisation
6	Context-free grammars, derivations and ambiguity
7	Chomsky normal form and the CYK algorithm
8	Pushdown automata and their equivalence with context-free grammars
9	Non-context-free languages; closure properties
10	Turing machines and their variants
11	The Church–Turing thesis; universal Turing machines
12	Decidability and the halting problem
13	Reductions; Rice's theorem; the Post correspondence problem
14	Time complexity and the class P
15	The class NP and the Cook–Levin theorem
16	NP-complete problems and reductions
17	Space complexity: PSPACE and Savitch's theorem
18	Course Review & Exam

5. Teaching and Learning Methods

The module is self-paced. All materials are provided on Moodle. Each unit has a lecture and exercise sheets with constructions and proofs.

Activity	Hours
Lectures	36
Labs / Tutorials	24
Independent study	140
Total	**200**

6. Assessment

Your knowledge will be assessed throughout the course with quizzes, exercise sheets and the final exam. The exam checks whether you can build automata and grammars for given languages, and whether you can write correct proofs of non-regularity, undecidability and NP-completeness.

Component	Weight	Due	LOs
Quizzes	20%	Throughout	LO1–LO5
Exercise sheets	30%	Throughout	LO1–LO5
Final exam	50%	End of module	LO1–LO5
Total	**100%**		

7. Reading List

Essential

- M. Sipser: *Introduction to the Theory of Computation*, 3rd ed., Cengage Learning, 2012.

Recommended

- J. E. Hopcroft, R. Motwani, J. D. Ullman: *Introduction to Automata Theory, Languages, and Computation*, 3rd ed., Pearson, 2006.
- D. C. Kozen: *Automata and Computability*, Springer, 1997.
- S. Arora, B. Barak: *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009.

8. Materials and Licence

Course materials are published under the Creative Commons Attribution-ShareAlike 4.0 licence unless stated otherwise. See wosm.academy/licence for details.

9. Policies

- **Attendance:** Attendance is not mandatory. The module is self-paced, so you can study the materials when it suits you.
- **Late submissions:** Work submitted after the deadline is penalised.
- **Academic integrity:** All submitted work must be your own. See the Code of Conduct.