GHENT UNIVERSITY

Course Specifications

From the academic year 2020-2021 up to and including the academic year

Automata, Computability and Complexity (C003785)

Course size *(nominal values; actual values may depend on programme)*

Credits 6.0	**Study time 180 h**	**Contact hrs** 60.0h

Course offerings and teaching methods in academic year 2022-2023

A (semester 2)	Dutch	Gent	online lecture	0.0h
			seminar: practical PC room classes	30.0h
			lecture	30.0h
			online seminar: coached exercises	0.0h

Lecturers in academic year 2022-2023

Storme, Leo	WE16	lecturer-in-charge

Offered in the following programmes in 2022-2023	**crdts**	**offering**
Bachelor of Science in Computer Science	6	A
Master of Science in Teaching in Science and Technology(main subject Mathematics)	6	A
Master of Science in Mathematics	6	A

Teaching languages

Dutch

Keywords

Finite automata, formal languages, push down automata, Turing machines, computability, complexity

Position of the course

This course concentrates on the topics formal language, automaton, computability, and complexity, and the interactions between these aspects of computer science, which permits to investigate problems about computer science, in order to obtain a better insight in computer science.
Many ideas have led to the present knowledge about computer science. The fact that the theory about computer science is based on solid ideas, can be seen in many ways. This includes the hierarchies regarding problems (languages) and automata (machines) which correspond to each other. We investigate which types of problems (languages) can be investigated by important types of automata. But we also pay attention to the fact that many problems cannot be solved by an automaton. Obtaining insight that certain, and which, problems are not solvable by an automaton is a central theme of this course.
But not only whether a problem can be solved by an automaton is important. In practice, also the complexity of a problem plays an important part. This is why we also pay attention to the complexity of problems, and we show that there are still many questions concerning the complexity of problems.

Contents

The topics of this course include:
1. finite automata and regular languages: (non-)determinism, pumping lemma, optimalization.
2. pushdown automata and context free languages: general properties, pumping lemma.
3. Turing machines and recursively enumerable languages: (non-)determinism, (semi)decidable problems, thesis of Church, halting problem, reduction of problems, theorem of Rice.

4. Complexity classes: classes P and NP, NP-complete problems, is P=NP?

Initial competences

Discrete mathematics and logic, including sets, functions, relations, recursion, matrices, graphs, and proof techniques. Data structures and algorithms. Programming in a high-level language.

Final competences

1 The student will know different descriptions of various classes of formal languages.
2 The student is able to prove regularity or context-freeness of languages. In addition he will also be able to prove non regularity or non context freeness of languages.
3 The student is able to define automata, pushdown automata and Turing machines which perform prescribed tasks.
4 The student knows basic unsolvable problems and will be able to prove unsolvability by means of reductions.
5 The student knows the basics of the complexity classes P and NP, and of NP-complete problems.

Conditions for credit contract

Access to this course unit via a credit contract is determined after successful competences assessment

Conditions for exam contract

This course unit cannot be taken via an exam contract

Teaching methods

Online lecture, Lecture, Online seminar: coached exercises, Seminar: practical pc room classes

Extra information on the teaching methods

Theory and exercises are presented during the lectures, and the students also have to make exercises, under supervision of the assistant.
Theory and exercises: because of COVID19, alternative didactic methods can be used when this is necessary.

Learning materials and price

Lecture notes based on the book
E. Rich, *Automata*, *Computability and Complexity: Theory and Applications.*
Prentice Hall (2007), ISBN 978-0-13-228806-4.
Slides are available via Ufora

References

1 A. Aho, R. Sethi, J.D. Ullman: *Compilers: Principles, Techniques and Tools.* Addison Wesley 1986.
2 J.E. Hopcroft, J.D. Ullman: *Introduction to Automata Theory, Languages and Computation*: Addison Wesley 1979.
3 D. Kozen, A*utomata and Computability*, Springer 1999.
4 P. Linz: *An Introduction to Formal Languages and Automata*, Fourth Edition. Sudbury, Mass.: Jones and Bartlett Publishers, 2006.
5 E. Rich, *Automata*, *Computability and Complexity: Theory and Applications*. Prentice Hall (2007).

Course content-related study coaching

Lecturer and assistant are available for the student to assist with problems on the course. An electronic learning environment allows easy communication between students and teachers.

Assessment moments

end-of-term assessment

Examination methods in case of periodic assessment during the first examination period

Open book examination, Written examination with open questions

Examination methods in case of periodic assessment during the second examination period

Open book examination, Written examination with open questions

Examination methods in case of permanent assessment

Possibilities of retake in case of permanent assessment

not applicable

Extra information on the examination methods

The exam (the theory as well as the exercises) is open book.

Calculation of the examination mark

First exam period: periodic evaluation (100%): theory 40% and exercises 60%.
Second exam period: periodic evaluation (100%): theory 40% and exercises 60%.