Course number	U-LAS30 20044 LE10		
Course title (and course title in English)	Introduction to Formal Languages-E2 Introduction to Formal Languages-E2	Instructor's name, job title, and department of affiliation	Graduate School of Informatics Program-Specific Associate Professor,Jesper Jansson
Group	Informatics	Field(Classification)	(Issues)
Language of instruction	English	Old group	
Number of credits	2		
Number of weekly time blocks	1	Class style	Lecture (Face-to-face course)
Year/semesters	2025・First semester		
Days and periods	Mon.1	Target year	All students
Eligible students	For all majors		

[Overview and purpose of the course]

Formal language theory is a fundamental area of theoretical computer science that studies (among other things) different ways of representing possibly infinite collections of words having some shared structure.
It is closely related to computability, computational complexity, and mathematical logic, and has practical applications in linguistics, artificial intelligence, and the design of programming languages.
The purpose of this course is to provide an introduction to formal language theory for non-computer science students.
The main topics include finite-state automata, regular languages, pushdown automata, context-free languages, Turing machines, and decidability.

[Course objectives]

After completing this course, the student should be able to:
- Explain the relationships between different classes of formal languages, automata, and grammars.
- Design an automaton or a grammar that accepts or generates a specified formal language, and conversely, determine the formal language that is accepted or generated by a specified automaton or grammar.
- Prove or disprove mathematical properties of formal languages, grammars, and automata.
- Use the diagonalization method or reductions to establish that certain languages are undecidable.
- Understand how the concept of "information" can be defined using computability theory.

[Course schedule and contents)]

The course will cover the following topics:
1. Introduction
2. Finite-state automata, regular languages, nondeterminism (1)
3. Finite-state automata, regular languages, nondeterminism (2)
4. Finite-state automata, regular languages, nondeterminism (3)
5. Finite-state automata, regular languages, nondeterminism (4)
6. Pushdown automata, context-free languages, grammars (1)
7. Pushdown automata, context-free languages, grammars (2)
8. Pushdown automata, context-free languages, grammars (3)
9. Turing machines (1)
10. Turing machines (2)
11. Decidability
12. Reducibility (1)
13. Reducibility (2)

Continue to Introduction to Formal Languages-E2(2)↓↓↓

Introduction to Formal Languages-E2(2)

14. Course summary and Q & A session
<<Final examination>>
15. Feedback

[Course requirements]

An ability to think abstractly and to solve problems of a mathematical nature will be required for this course. No programming skills are needed.

[Evaluation methods and policy]

A written examination at the end of the course.

[Textbooks]

M. Sipser 『Introduction to the Theory of Computation, Third Edition』 （Cengage Learning）ISBN:978-1133187790（2012）

[Study outside of class (preparation and review)]

Students will be expected to spend about 3 hours per week to prepare for and review the lessons.

[Other information (office hours, etc.)]