课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	**课程名称 Course Title**	**Introduction to Theory of Computation 计算理论导论**
2.	**授课院系** **Originating Department**	计算机科学与工程系 Department of Computer Science and Engineering
3.	**课程编号** **Course Code**	CS338
4.	**课程学分 Credit Value**	3
5.	**课程类别** **Course Type**	专业选修课 Major Elective Courses
6.	**授课学期** **Semester**	春季 Spring
7.	**授课语言** **Teaching Language**	英文 English
8.	**授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师）** **Instructor(s), Affiliation& Contact** **（For team teaching, please list all instructors）**	Pietro Simone Oliveto，Professor，计算机科学与工程系，olivetop@sustech.edu.cn Pietro Simone Oliveto, Professor, Department of Computer Science and Engineering, t olivetop@sustech.edu.cn
9.	**实验员/助教、所属学系、联系方式** **Tutor/TA(s), Contact**	无 NA / 待公布 To be announced / 已确定的实验员/助教联系方式 Please list all Tutor/TA(s) **（请保留相应选项 Please only keep the relevant information）**
10.	**选课人数限额(可不填)** **Maximum Enrolment（Optional）**	

		讲授 Lectures	习题/辅导/讨论 Tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify）	总学时 Total
11.	授课方式 Delivery Method 学时数 Credit Hours	32		32		64

12.	先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	CS203 Data Structures and Algorithm Analysis
13.	后续课程、其它学习规划 Courses for which this course is a pre-requisite	
14.	其它要求修读本课程的学系 Cross-listing Dept.	

教学大纲及教学日历 SYLLABUS

15. 教学目标 Course Objectives

This course introduces the theoretical foundations of computer science. The first part of the course focuses on computability theory i.e., "what is computable". Students will learn the notions of computation and appreciate what can and what cannot be computed in principle. Computational models of increasing complexity will be rigorously defined and analyzed to identify the classes of problems that the respective models can solve and which classes of problems they cannot solve. Students will learn to appreciate to what extent an increase in computational power allows a computational model to solve larger classes of problems, and what problems, instead, are not solvable by any reasonable computational model independent of the amount of finite available resources. The second part of the course focuses on complexity theory i.e., "what is computable efficiently". Students will learn what classes of problems computational models can solve when provided with reasonable finite resources (time and space) and which classes of problems cannot be solved efficiently by any reasonable computational model. During the course students will encounter some of the most important open problems in computer science. Students will also learn to appreciate why the notions of optimality are relaxed to seeking approximate solutions and fixed parameter tractable solutions, rather than optimal ones, in the advanced algorithm design courses they study throughout their degree program. Overall, the course provides a thorough grounding into the theoretical foundations of computer science and prepares students for more advanced theory courses.

16. 预达学习成果 Learning Outcomes

On completion of this course, students will be able to:

- Analyse rigorously whether a computational problem is solvable or not by different computational models, including finite state machines, pushdown automata, and Turing machines.
- Describe the classes of problems that different models of computation can compute solvable, and derive whether a given problem belongs to the class of those solvable by a given model or not.
- Provide examples of problems that no reasonable computational model can solve independent of the amount of available finite resources, and be capable of showing when a problem is at least as hard as one that is not computable in finite time.
- Show whether a computational problem can be solved efficiently or whether it cannot given reasonable

computational resources.

- Be able to prove to which computational complexity class a problem belongs to (i.e., how hard the problem is).

17. 课程内容及教学日历 （如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）
Course Contents (in Parts/Chapters/Sections/Weeks. Please notify name of instructor for course section(s), if this is a team teaching or module course.)

16 2-hour lectures and 16 2-hour Labs. In the Lab sessions the students will carry out exercises to reinforce the understanding of the concepts introduced at lecture.

PART A – Computability Theory

1. Finite Automata (DFA) & Regular Languages
2. Non Deterministic Finite Automata (NFA)
3. Regular Expressions
4. Non Regular Languages
5. Pushdown Automata
6. Context Free Languages
7. Non Context Free Languages
8. Turing Machines
9. Variants of Turing Machines
10. Decidability & recognizability
11. Undecidability
12. Reducibility

PART B – Complexity Theory

13. P Class
14. NP Class
15. NP Completeness
16. Cook-Levin Theorem (SAT is NP-Complete)

18. 教材及其它参考资料 **Textbook and Supplementary Readings**

Micheal Sipser. Introduction to the Theory of Computation. Cengage Learning, 2013 (3rd edition)

课程评估 ASSESSMENT

19.

评估形式 Type of Assessment	评估时间 Time	占考试总成绩百分比 % of final score	违纪处罚 Penalty	备注 Notes
出勤 Attendance				
课堂表现 Class Performance				
小测验 Quiz				
课程项目 Projects				
平时作业 Assignments		40%		
期中考试 Mid-Term Test				
期末考试 Final Exam		60%		
期末报告 Final Presentation				
其它（可根据需要改写以上评估方式） Others (The above may be modified as necessary)				

20. 记分方式 **GRADING SYSTEM**

☑A. 十三级等级制 **Letter Grading**
☐ B. 二级记分制（通过/不通过） **Pass/Fail Grading**

课程审批 REVIEW AND APPROVAL

21. 本课程设置已经过以下责任人/委员会审议通过
This Course has been approved by the following person or committee of authority