The Theory of Computation

(Guidance)

Jingde Cheng

Southern University of Science and Technology

CS327: The Theory of Computation (ToC)

♣ Teaching professor

- Jingde Cheng (程京德), chengjd@sustech.edu.cn
- HP: http://www.aise.ics.saitama-u.ac.jp/~cheng/index.html
- Blog: http://blog.sciencenet.cn/u/JingdeCheng (科学网博客：计算机科学，数理逻辑，人工智能，等等)
- Use Google or 百度 to find more information about me.

♣ Teaching assistant teacher

- Weiyu Wang (王维语), wangwy@mail.sustech.edu.cn
- Answering class (optional): Will be announced.

♣ QQ group

- CS327-ToC: **974628546**

The Theory of Computation: What Is It and Why Study It?

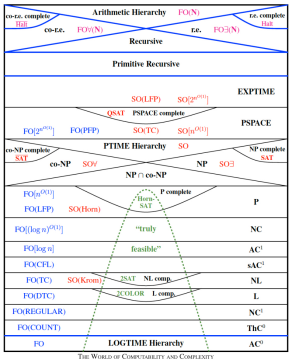
♣ The Theory of Computation (ToC): What is it?

- The ToC studies the nature of computation and various properties of computation.
- It is the core theoretical foundation of Computer Science (CS), Intelligent Science (IS), and Artificial Intelligence (AI).

♣ The Theory of Computation (ToC): Why study it?

- The ToC will let you know:
  - What can and cannot be computed in principle.
  - The computational time complexity, the computational space complexity, and the classification of computational difficulty.
- You cannot be regarded as a professional in any area of CS, IS, and AI, if you have not studied the ToC.

The World of Computability and Complexity [SEP]



The World of Computability and Complexity

Inclusion Relationships among Major Complexity Classes [SEP]

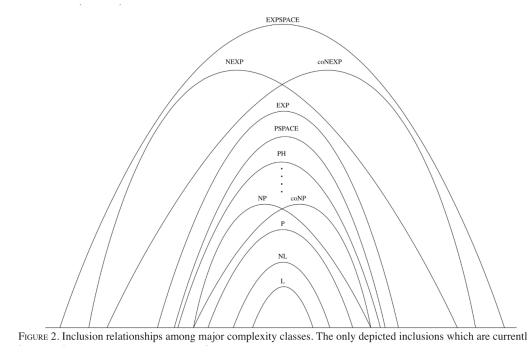


FIGURE 2. Inclusion relationships among major complexity classes. The only depicted inclusions which are currently known to be proper are **L** ⊊ **PSPACE** and **P** ⊊ **EXP**.

CS327: The Theory of Computation

- Guidance
- Enumerability and Diagonalization
- Finite Automata and Regular Languages
- Context-Free Languages
- Computation: Turing Machines
- Computation: Turing-Computability (Turing-Decidability)
- Computation: Reducibility (Turing-Reducibility)
- Computation: Recursive Functions
- Computation: Recursive Sets and Relations
- Equivalent Definitions of Computability
- Advanced Topics in Computability Theory
- Computational Complexity
- Time Complexity
- Space Complexity
- Intractability
- Advanced Topics in Complexity Theory

Major Text Books

- M. Sipser, "Introduction to the Theory of Computation," Cengage Learning, 2013 (3rd Edition).
- G. S. Boolos, J. P. Burgess, and R. C. Jeffrey, "Computability and Logic," Cambridge University Press, 2007 (5th Edition).
- P. Linz, "An Introduction to Formal Languages and Automata," Jones & Bartlett Learning, 2017 (6th Edition).

Other Text and Reference Books

- R. Weber, "Computability Theory," AMS, 2012. [W-12]
- C. Tourlakis, "Theory of Computation," Wiley, 2012. [T-12]
- J. C. Martin, "Introduction to Languages and the Theory of Computation," McGraw-Hill, 2011 (4th Edition). [M-11]
- S. Homer and A. L. Selman, "Computability and Complexity Theory," Springer, 2011 (2nd Edition). [HS-11]
- C. Moore and S. Mertens, "The Nature of Computation," Oxford University Press, 2011. [MM-11]
- M. Fernandez, "Models of Computation -- An Introduction to Computability Theory," Springer, 2009. [F-09]
- O. Goldreich, "Computational Complexity – A Conceptual Perspective," Cambridge University Press, 2008. [G-08]
- J. R. Hindley and J. P. Seldin, "Lambda-Calculus and Combinators, An Introduction," Cambridge University Press, 2008. [HS-08]
- J. E. Hopcroft, R. Motwani, and J. D. Ullman, "Introduction to Automata Theory, Languages, and Computation," Pearson Education, 2007 (3rd Edition). [HMU-07]

Other Text and Reference Books

- H. R. Lewis and C. H. Papadimitriou, "Elements of the Theory of Computation," Prentice-Hall, 1998 (2nd Edition). [LP-98]
- D. S. Bridges, "Computability – A Mathematical Sketchbook," Springer, 1994. [B-94]
- C. H. Papadimitriou, "Computational Complexity," Addison-Wesley, 1994. [P-94]
- P. Odifreddi, "Classical Recursion Theory – The Theory of Functions and Sets of Natural Numbers," Elsevier, 1992. [O-92]
- N. Cutland, "Computability – An Introduction to Recursive Function Theory," Cambridge University Press, 1980. [C-80]
- M. R. Garey and D. S. Johnson, "Computers and Intractability – A Guide to the Theory of NP-Completeness," 1979. [GJ-79]
- M. Davis, "Computability and Unsolvability," McGraw-Hill, 1958. [D-58]
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms," 2009 (3rd Edition). [CLRS-I2A-09]
- M. Davis, "Engines of Logic: Mathematicians and the Origin of the Computer," Norton, 2000. [D-EoL-00]

CS327: The Theory of Computation

♣ Pre-requisites

- CS101(A/B): Introduction to Computer Science (A/B)
- CS104/CS108: Introduction to Mathematical Logic
- Those students who haven't learned the naive Set Theory and Mathematical Logic should learn them by themselves in advance.

♣ Notes

- Because of time limit, I do not answer any question about the naive Set Theory and Mathematical Logic in classes and QQ group.
- My courseware for the naive Set Theory and Mathematical Logic (and some other reading material) can be found in QQ group.

My Hope/Statement and Your Attitude

♣ My hope/statement

- I really like to teach and guide you to become world-class scientists and/or engineers in the future, by laying a good foundation for you.

♣ Your attitude

- (I want you) to listen carefully, think deeply, ask questions actively, and read text books and reference materials carefully.

♣ "青出于蓝而胜于蓝"

- "青，取之于蓝，而胜于蓝；冰，水为之，而寒于水。" -- 荀子, "劝学", 约公元前230年
- 我将把你们视为清华的学生来教授你们知识和能力(但是按照南科大学生要求)；期待着你们"青出于蓝而胜于蓝"。

My Teaching Style

♣ For important notions/concepts

- What is it?
- Why study it?

♣ For important assumptions and/or principles

- Why ToC needs it?
- Why it is fundamental? / What is underlain by it?

♣ For important problems

- What is its background?
- Why it is interesting?

♣ Using QQ group

- Send various advices/materials/notices to you, and answer your questions.

Some Rules You Should Follow

- **Preview**
  - I will distribute my courseware and some reading materials at least one week before class every week.
  - You should preview my courseware and the reading materials before class every week.
  - Note: It is very difficult to understand all the teaching contents at once just listening in class !!!
- **Ask questions in class but not break time**
  - I will give you time to ask questions in class.
  - You should ask questions in class, but not break time, because my answers in class are beneficial to all students.

Assessment Grading

- **Homework (50% of final score, 100/130 points)**
  - Answer questions, read materials, and submit reports
  - You can (should!) get full points (100) if you do homework hard, and I will let you pass this course if you get full points of homework.
  - Note: **The amount of homework is large !!!**
- **Final examination (50% of final score, 100/120 points)**
  - Open-book examination
  - **Do not consider that the open-book examination is easy !!!**
  - Only those students who have studied and reviewed the course contents very seriously can get full points (100).

Introduction to the Theory of Computation

- Enumerability and Diagonalization
- Finite Automata and Regular Languages
- Context-Free Languages
- Computation: Turing Machines
- Computation: Turing-Computability (Turing-Decidability)
- Computation: Reducibility (Turing-Reducibility)
- Computation: Recursive Functions
- Computation: Recursive Sets and Relations
- Equivalent Definitions of Computability
- Advanced Topics in Computability Theory
- Computational Complexity
- Time Complexity
- Space Complexity
- Intractability
- Advanced Topics in Complexity Theory