

CS 3530: Computational Theory

Resources

- [Syllabus](#)
-

Introduction and Mathematical Preliminaries

August 22

[\[slides\]](#)

Topics

- Introduction to Formal Systems and Computation [no reading]
- Mathematical Preliminaries [Sipser §0.1 and §0.2]
- Doing Proofs [Sipser §0.3 and §0.4]
- Introduction to LaTeX [no reading]

Resources:

- [How to write proofs](#)
- [The Greek alphabet](#)
- [Setting up LaTeX](#)
- [LaTeX Book](#)
- [LaTeX Documentation](#)
- [LaTeX Cheat Sheet](#)
- [Tool to find the LaTeX name for a symbol](#)
- [LaTeX formula from handwriting](#)
- [Metapost macros for making finite automata figures in LaTeX \(you can download the theory.mp file directly here\)](#)

To install LaTeX on Linux:

```
sudo apt-get install texlive-latex-recommended texlive-metapost texlive-fonts-recommended
```

To install on macOS, install MacTeX:

- <https://www.tug.org/mactex/mactex-download.html>

To install on Windows, install TeXLive:

- <https://www.tug.org/texlive/acquire-netinstall.html>

To compile documents, make sure `theory.mp` is in the same directory, then run:

```
pdflatex -shell-escape inputfile
pdflatex -shell-escape inputfile
```

i.e., run it twice. Or you may modify and use this Makefile:

- [Makefile for LaTeX](#)

Just assign the list of input `.tex` files you want compiled to the `SRC=` line and run `make`. To make a specific file, even if it is not in the `SRC` line, use `make file.pdf`. When I will be working on a file for a while, I like to change the `all` rule to point to the file I am currently trying to create, then all I have to type is `make` to build it.

Finite Automata and Regular Languages

September 5

[\[slides\]](#)

Topics

- Finite Automata [Sipser §1.1 and §1.2]
- NFAs and DFAs Closure Properties [Sipser §1.2]
- Regular Expressions [Sipser §1.3]
- Countability [Sipser, “The Diagonalization Method,” pages 174-178 (from just before Definition 4.12 up to Corollary 4.18)]
- Non-regular Languages [Sipser §1.4]

Assignments

- Assignment 1: Finite Automata ([1a pdf](#), [1b](#), [1c](#))
 - Assignment 2: Regular Expressions ([2a](#), [2b](#))
 - Assignment 3: Regular Languages ([3a](#), [3b](#))
-

Context-free Languages

September 21

[\[slides\]](#)

Topics

- Context-Free Grammars [Sipser §2.1 (except Chomsky Normal Form)]
- Pushdown Automata [Sipser §2.2]
- CFL Closure Properties and Non-Context-Free Languages [Sipser §2.3]
- Context-Free Recognition [Sipser §2.1 (Chomsky Normal Form)]

Assignments

- Assignment 4: Context-free Grammars and Pushdown Automata ([4a](#), [4b](#), [4c](#))
 - Assignment 5: Context-free Languages ([5a](#), [5b](#))
-

Midterm Exam

- Review on October 5
 - Midterm exam on October 10
-

Computability

October 17

[\[slides\]](#)

Topics

- Turing Machines [Sipser §3.1]
- Church’s Thesis [Sipser §3.2, §3.3]
- Decidability and a Universal Turing Machine [Sipser §4.1]

Assignments

- Assignment 6: Turing Machines ([6a](#), [6b](#), [6c](#), [6d](#))
-

Uncomputability

October 31

[\[slides\]](#)

Topics

- Undecidability [Sipser §4.2, §5.1]
- Reductions and Rice's Theorem [Sipser Ch. 5]
- Undecidable Problems and Unprovable Theorems [Sipser Ch. 5]

Assignments

- Assignment 7: Computability ([7a](#), [7b](#), [7c](#), [7d](#))

Resources:

- [The halting problem as a poem](#)
 - [Busy beaver problem](#)
-

Computational Complexity

November 14

[\[slides\]](#)

Topics

- Computational Complexity [Sipser §7.1, §7.2]
- Polynomial Time [Sipser §7.2]
- NP [Sipser §7.3]
- NP-Completeness [Sipser §7.4, §7.5]
- Cook-Levin Theorem and Beyond [no reading]
- Conclusions [no reading]

Assignments

- Assignment 8: Complexity ([8a](#), [8b](#), [8c](#), [8d](#))

Resources:

- [A summary of how to write NP-completeness proofs](#)
-

Final Exam

- Review on December 6
- Final exam on Tuesday, December 12 at 9:00 am