# Independent Study
# Complexity Theory

Ryan Dougherty

# Table of Contents

## 1 Introduction & Preface

Welcome to this series of lecture notes! The main book that the material comes from is Arora and Barak's *Computational Complexity* book [AB09]. Some material that is assumed from the reader (and is referenced in Section 2) is from Sipser's *Introduction to the Theory of Computation* book [Sip12]. We assume that the reader has a reasonable understanding of the following material:

- {Regular, Context-free, Turing-decidable, Turing-recognizable} languages, and their machine counterparts
- (Un)decidability
- Reducibility
- Recursion theorem
- Time complexity: $\mathcal{P}, \mathcal{NP}, \mathcal{EXPTIME}$, and their -complete versions
- Space complexity: $\mathcal{PSPACE}, \mathcal{EXPSPACE}, \mathcal{L}, \mathcal{NL}$, and their -complete versions

## 2 Review

This section highlights many of the key definitions and theorems studied in a first-year graduate (or advanced undergraduate) course in complexity theory. We assume the reader knows about finite automata (DFAs/NFAs), grammars (CFGs), and Turing machines (TMs), and their respective language classes.

**Definition 1.** *A TM is a* decider *if it halts (accepts or rejects) on every input.*

**Definition 2.** *A language $B$ is* decidable *if there exists a decider $D$ such that $L(D) = B$. A language $C$ is* undecidable *if $C$ is not decidable.*

**Theorem 1.** *The following are decidable:*

- $A_{DFA} = \{\langle M, w\rangle : M \textit{ is a DFA that accepts } w\}$.
- $E_{DFA} = \{\langle M\rangle : M \textit{ is a DFA whose language is empty}\}$.
- $ALL_{DFA} = \{\langle M\rangle : M \textit{ is a DFA whose language is } \Sigma^*\}$.
- $EQ_{DFA} = \{\langle M_1, M_2\rangle : M_1 \textit{ and } M_2 \textit{ are DFAs and } L(M_1) = L(M_2)\}$.
- $A_{CFG} = \{\langle G, w\rangle : G \textit{ is a CFG that generates } w\}$.
- $E_{CFG} = \{\langle G\rangle : L(G) \textit{ is empty}\}$.

**Theorem 2.** *The following are undecidable:*

- $ALL_{CFG} = \{\langle G\rangle : G \textit{ is a CFG and } L(G) = \Sigma^*\}$.
- $EQ_{CFG} = \{\langle G_1, G_2\rangle : G_1 \textit{ and } G_2 \textit{ are CFGs and } L(G_1) = L(G_2)\}$.
- $A_{TM} = \{\langle M, w\rangle : M \textit{ is a TM that accepts } w\}$.

## 3 Polynomial Hierarchy, Alternating TMs

# 4 Boolean Circuits

# 5 Randomization

# 6 Interactive Proofs

# 7 Quantum Computation

# 8 PCP Theorem

## 9 Decision Trees

# 10 Communication Complexity

# 11 Algebraic Computation Models

## 12 Counting Complexity

## 13 Average-Case Complexity

# 14 Hardness Amplification

## 15 Derandomization

# 16 Expanders/Extractors

## 17 PCP and Fourier Transform

## 18 Parameterized Complexity

## References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[Sip12] Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, 2012.