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.

2.1 (Un)Decidability

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:*

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

Theorem 2. *The following are undecidable:*

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

Theorem 3. *The class of decidable languages is closed under complement.*

Definition 3. *A language B is* Turing-recognizable *(or recognizable) if there exists a TM that recognizes B. A language C is co-Turing-recognizable (or co-recognizable) if it is the complement of some Turing-recognizable language.*

Theorem 4. *A_{TM} is not co-recognizable.*

Theorem 5. *A language B is decidable if and only if B is recognizable and co-recognizable.*

Definition 4. *A function $f : \Sigma^* \to \Sigma^*$ is a* computable function *if there exists a TM that, on input w, halts with $f(w)$ on its tape.*

2.2 Reducibility

Definition 5. *A language A is* mapping-reducible *to language B, written $A \leq_m B$, if there exists a computable function such that $w \in A$ if and only if $f(w) \in B$.*

Theorem 6. *If $A \leq_m B$ and B is decidable, then A is decidable; if A is undecidable, then B is undecidable.*

Corollary 1. *$HALT_{TM} = \{\langle M, w\rangle : M \text{ is a TM that halts on input } w\}$ is undecidable.*

Definition 6. *A TM's language has a* property *P (a subset of all TM descriptions) such that whenever M_1, M_2 are TMs, and $L(M_1) = L(M_2)$, $\langle M_1\rangle \in P$ if and only if $\langle M_2\rangle \in P$. A property P is* nontrivial *if some TM has property P and some other TM does not.*

(Rice's Theorem) 7. *Deciding whether a TM has a nontrivial property P of its language is undecidable.*

Theorem 8. *$EQ_{TM} = \{\langle M_1, M_2\rangle : M_1, M_2 \text{ are TMs and } L(M_1) = L(M_2)\}$ is undecidable; also, it is neither recognizable nor co-recognizable.*

Definition 7. *A* configuration *of a TM on input* $w = w_1 \cdots w_n$ *in state* q *is* $w_1 \cdots w_{i-1} q w_i \cdots w_n$*. A* computation history *is a set of configurations delimited by an extra symbol #:* $\#C_1\#C_2\# \cdots \#C_\ell\#$*, where* C_i *logically yields* C_{i+1}*. An* accepting computation history *is one such that* C_1 *is the start configuration, and* C_ℓ *is an accepting one.*

Definition 8. *A* linear bounded automaton *(LBA) is a TM that does not allow to move the tape head past the right end of the input.*

Theorem 9. $A_{LBA} = \{\langle M, w\rangle : M$ *is an LBA that accepts* $w\}$ *is decidable.*

Definition 9. *The* Post Correspondence Problem *(PCP) is a puzzle, with a given set of tiles with nonempty "top strings" and nonempty "bottom strings." The objective is to list the tiles, repetitions allowed, such that the concatenation of the top strings of all the chosen tiles equals the same of the bottom strings.*

Theorem 10. *PCP is undecidable.*

(Recursion Theorem) 11. *Let a TM* T *compute a function* $t : \Sigma^* \times \Sigma^* \to \Sigma^*$*. Therefore, there exists a TM* R *that computes a function* $r : \Sigma^* \to \Sigma^*$*, such that* $r(w) = t(\langle R\rangle, w)$ *for all* w*. In other words, every TM can obtain their own description.*

Definition 10. *A TM* M *is* minimal *if there does not exist a TM* N *that has fewer states and* $L(M) = L(N)$*.*

Theorem 12. $MIN_{TM} = \{\langle M\rangle : M$ *is a TM and has a minimal number of states*$\}$ *is not recognizable.*

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.