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
- Space complexity

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. This review roughly covers the first four chapters of [AB09], and the first eight (and part of the ninth) chapters of [Sip12].

2.1 (Un)Decidability

Definition 1. *A TM is a* decider *if it halts (accepts or rejects) on every input. A language* B *is* decidable *if there exists a decider* D *such that* $\mathrm{L(D)} = \mathrm{B}$*. A language* C *is* undecidable *if* C *is not decidable.*

Theorem 1. *The following are decidable:*

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

Theorem 2. *The following are undecidable:*

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

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

Definition 2. *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. $\mathrm{A_{TM}}$ *is not co-recognizable.*

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

2.2 Reducibility

Definition 3. *A function* $\mathrm{f} : \Sigma^* \to \Sigma^*$ *is a* computable function *if there exists a TM that, on input* w*, halts with* $\mathrm{f(w)}$ *on its tape. A language* A *is* mapping-reducible *to language* B*, written* $\mathrm{A} \leq_{\mathrm{m}} \mathrm{B}$*, if there exists a computable function* f *such that* $\mathrm{w} \in \mathrm{A}$ *if and only if* $\mathrm{f(w)} \in \mathrm{B}$.

Theorem 6. *If* $\mathrm{A} \leq_{\mathrm{m}} \mathrm{B}$ *and* B *is decidable, then* A *is decidable; if* A *is undecidable, then* B *is undecidable; if* B *is recognizable, then* A *is recognizable; if* A *is not recognizable, then* B *is not recognizable.*

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

Definition 4. *A TM's language has a* property P *(a subset of all TM descriptions) such that whenever* $\mathrm{M_1}, \mathrm{M_2}$ *are TMs, and* $\mathrm{L(M_1)} = \mathrm{L(M_2)}$*,* $\langle \mathrm{M_1}\rangle \in \mathrm{P}$ *if and only if* $\langle \mathrm{M_2}\rangle \in \mathrm{P}$*. A property* P *is* nontrivial *if some TM has property* P *and some other TM does not.*

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

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

Definition 5. *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 6. *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 7. *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.*

Theorem 11 (Recursion Theorem). *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 8. *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 is minimal*$\}$ *is not recognizable.*

2.3 Logical Theories

Definition 9. *A formula over some operations is* atomic *if* R_i *over variables* $x_1, \cdots, x_\ell$ *is a relation of arity* ℓ*. A formula* φ *is* well-formed *if it is atomic, a formula formed from other atomic formulas using the operations, or of the form* $\exists x[\varphi_1]$ *or* $\forall x[\varphi_1]$ *where* φ_1 *is a well-formed formula. A variable is* bounded *if it is within the scope of a quantifier, and free otherwise. A well-formed formula with no free variables is a* sentence *or a* statement*. The* universe *is the set of possible values for each variable, and the* model *specifies the universe and relations used. The* theory *of a model* M*, called* $Th(M)$*, is the set of true statements. A formula in* prenex normal form *is one that has all quantifiers appear first.*

Theorem 13. $Th(\mathbb{N}, +)$ *is decidable.*

Theorem 14. $Th(\mathbb{N}, +, \times)$ *is undecidable.*

Definition 10. *A* formal proof *of a statement* φ *is a sequence of statements* $S_1, \cdots, S_\ell$ *where* $S_\ell = \varphi$*, where each* S_i *follows logically from preceding statements and* axioms *(statements not requiring a proof). If all provable statements are true, then the system is* sound*; if all true statements are provable, then the system is* complete*.*

Theorem 15. $Provable(\mathbb{N}, +, \times) = \{$*set of statements in* $(\mathbb{N}, +, \times)$ *that have proofs*$\}$ *is recognizable.*

Theorem 16. *There exists a true, but unprovable statement in* $Th(\mathbb{N}, +, \times)$*.*

2.4 Oracle TMs

Definition 11. *An* oracle TM M *is a TM with an "oracle tape" that, when the TM writes a string onto this tape, invokes the oracle (of a language* L*) and decides membership of the written string in* L *in zero time, and returns a yes or no answer (written as* M^L*). A language* A *is* decidable relative to *a language* B*–*$A \leq_T B$*–if there is a TM* M^B *that decides* A. A *is* Turing-reducible *to* B *if and only if* $A \leq_T B$.

Theorem 17. $E_{TM} \leq_T A_{TM}$.

Theorem 18. $A'_{TM} = \{\langle M, w\rangle : M$ *is a TM with an oracle for* A_{TM} *and* M *accepts* $w\}$ *is undecidable relative to* A_{TM}.

Definition 12. *The* minimal description *of a string* x *(*$d(x)$*)is the shortest string* $\langle M, w\rangle$ *where TM* M*, on input* w*, halts with* x *on the tape. The* descriptive complexity *of* x *(*$K(x)$*) is* $|d(x)|$

Theorem 19. $K(x) \leq |x| + c$ *for a constant* c.

Theorem 20. $K(xx) \leq |x| + d$ *for a constant* d.

Theorem 21. $K(xy) \leq 2\log_2(K(x)) + K(x) + K(y) + e$ *for a constant* e.

Definition 13. *A string* x *is* incompressible *if* $K(x) \geq |x|$.

Theorem 22. *At least half of all strings of length* $\leq n$ *are incompressible.*

Theorem 23. $K(x)$ *is not computable.*

Theorem 24. *No infinite subset of the set of incompressible strings is recognizable.*

2.5 Computational Complexity

Definition 14. $TIME(f(n))$ *(*$NTIME(f(n))$*) is the set of languages decidable within* $O(f(n))$ *steps on a single-tape deterministic (nondeterministic) TM.* $\mathsf{P} = \bigcup_{k\geq 0} TIME(n^k)$, $\mathsf{NP} = \bigcup_{k\geq 0} NTIME(n^k)$*. Decision on a NTM has that every computation branch halts, time is the number of transitions on the longest computation path, and space is the maximum number of cells visited on any computation path.*

Theorem 25. *The following are members of* p*:*

- *All regular languages*
- *All context-free languages*
- $PATH = \{\langle G, s, t\rangle : G$ *is an undirected graph having a path from* s *to* $t\}$

Definition 15. *A* verifier *is a TM that accepts a string* w *and a* certificate c*, and verifies whether* c *is valid.*

Theorem 26. NP*can also be defined as the set of languages with a polynomial-time verifier.*

Definition 16. *A language* A *is* polynomial-time reducible *to a language* B–$A \leq_p B$–*if the reduction takes polynomial time.*

Theorem 27. *Suppose* $A \leq_p B$*. If* $B \in \mathsf{P}$*, then* $A \in \mathsf{P}$*; if* $A \notin \mathsf{P}$*, then* $B \notin \mathsf{P}$.

Definition 17. *A boolean formula* φ *in* conjunctive normal form *is one that is a conjunction of clauses, and each clause is a disjunction of literals. A formula in* 3CNF *has* ≤ 3 *literals per clause. A formula is* satisfiable *if there exists an assignment to the variables to make the formula true.*

Theorem 28. *3SAT* $= \{\langle\varphi\rangle : \varphi$ *is a 3CNF formula that is satisfiable*$\} \in \mathsf{NP}$*, and 3SAT* $\leq_p$ *CLIQUE* $= \{\langle G, k\rangle : G$ *is a graph with a* k*-clique*$\}$.

Definition 18. *A language* B *is* NP-complete *if* $B \in \mathsf{NP}$*, and for every* $A \in \mathsf{NP}$*,* $A \leq_p B$*. If only the second condition is true, then* B *is* NP-hard.

Theorem 29. *The following are* NP*-complete:*

- *3SAT (the Cook-Levin theorem)*
- *CLIQUE*
- *INDSET (same as CLIQUE but no edges between vertices)*
- *VERTEX COVER (whether there exists a subset of vertices of size* $\leq k$ *such that every edge involves a vertex in the subset)*
- *HAMPATH (whether a directed graph contains a directed path through every vertex exactly once)*
- *UHAMPATH (undirected version of HAMPATH)*

2.6 Space Complexity

Definition 19. PSPACE$= \bigcup_{k\geq 0}\mathrm{SPACE}(n^k)$, NPSPACE$= \bigcup_{k\geq 0}\mathrm{NSPACE}(n^k)$.

Theorem 30. *For* $f(n) \geq n$, $\mathrm{TIME}(f(n)) \subseteq \mathrm{SPACE}(f(n))$.

Theorem 31 (Savitch's Theorem). *For* $f(n) \geq n$, $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$.

Corollary 2. PSPACE= NPSPACE.

Definition 20. *A language* B *Is* PSPACE-complete *if* $B \in$ PSPACE*, and for every* $A \in$ PSPACE, $A \leq_p B$. *If only the second condition is true, then* B *is* PSPACE*-hard.*

Theorem 32. *The following are* PSPACE*-complete:*

- *TQBF* $= \{\langle\psi\rangle : \psi$ *is a true quantified boolean formula*$\}$ *(i.e., of the form* $\psi = Q_1 x_1 \cdots Q_n x_n \varphi(x_1, \cdots, x_n)$ *where the* $Q_i \in \{\exists, \forall\}$*).*
- *FORMULA-GAME (a 2-player version of TQBF, where players take turns choosing values for the variables in order)*
- *Generalized Geography (a directed graph where each vertex is a string, and each edge has the next string start with the same letter as the previous one)*
- A_{LBA}

Definition 21. L$= \mathrm{SPACE}(\log(n))$, NL$= \mathrm{NSPACE}(\log(n))$.

Definition 22. *A* log-space transducer *is a deterministic TM with read-only input, write-only output, and a read-write work tape, and the space it uses is equal to the length of the non-blank portion of the work tape + log(size of input) + log(size of output). A language* A *is* log-space reducible *to a language* B *if there is a log-space transducer that computes a function* f *for which* $w \in B$ *if and only if* $f(w) \in A$*. A language* B *Is* NL-complete *if* $B \in$ NL*, and for every* $A \in$ NL, $A \leq_L B$*. If only the second condition is true, then* B *is* NL*-hard.*

Theorem 33 (Immerman-Szelepcsényi Theorem). *PATH* $= \{\langle G, s, t\rangle : G$ *is a directed graph with a directed* $s-t$ *path*$\}$ *is* NL*-complete, and* coNL*-complete.*

Corollary 3. NL= coNL.

Definition 23. *A function* f *is* space-constructible *if there is a TM that computes the function mapping* 1^n *(in unary) to* $f(n)$ *(in binary) in* $O(f(n))$ *space.*

Theorem 34 (Space Hierarchy Theorem). *If* f *is a space-constructible function, then there exists a language that can be decided in* $O(f(n))$ *space, but not in* $o(f(n))$ *space.*

Corollary 4. PSPACE$\neq$ EXPSPACE.

Corollary 5. NL$\neq$ PSPACE.

Definition 24. *A function* f*, which is* $\Omega(n\log(n))$*, is* time-constructible *if there is a TM that computes the function mapping* 1^n *(in unary) to* $f(n)$ *(in binary) in* $O(\frac{f(n)}{\log(n)})$ *time.*

Theorem 35 (Time Hierarchy Theorem). *If* f *is a time-constructible function, then there exists a language that can be decided in* $O(f(n))$ *time, but not in* $o(\frac{f(n)}{\log(f(n))})$ *time.*

Corollary 6. P$\neq$ EXP.

Corollary 7. *For any* $1 < c < d$, $\mathrm{TIME}(n^c) \neq \mathrm{TIME}(n^d)$.

Definition 25. *A language* B *Is* EXPSPACE-complete *if* $B \in$ EXPSPACE*, and for every* $A \in$ EXPSPACE, $A \leq_p B$*. If only the second condition is true, then* B *is* EXPSPACE*-hard.*

Theorem 36. $EQ_{REX\uparrow} = \{$*regular expressions with exponentiation*$\}$ *is* EXPSPACE*-complete.*

2.7 Relativized Complexity

Definition 26. P^A *(resp.,* NP^A*)* $= \{L : L$ *is decided by an oracle TM with an oracle for* A *in deterministic (nondeterministic) polynomial time*$\}$.

Theorem 37. *There exist oracles* A, B *such that* $\mathsf{P}^A = \mathsf{NP}^A$*, and* $\mathsf{P}^B \neq \mathsf{NP}^B$.

3 Polynomial Hierarchy, Alternating TMs

From Theorem 37, we have a notion of using P and NP with the power of oracle machines. However, we don't have a generalization of a "hierarchy" of such oracle machines (the theorem only concerns the "first level"). Therefore, in [MS72], the notion of a "polynomial hierarchy" was created. The hierarchy is defined (equivalently) as follows:

- $\Delta_0^{\mathsf{P}} = \Sigma_0^{\mathsf{P}} = \Pi_0^{\mathsf{P}} = \mathsf{P}$,
- $\Delta_i^{\mathsf{P}} = \mathsf{P}^{\Sigma_{i-1}^{\mathsf{P}}}$,
- $\Sigma_i^{\mathsf{P}} = \mathsf{NP}^{\Sigma_{i-1}^{\mathsf{P}}}$,
- $\Pi_i^{\mathsf{P}} = \mathsf{coNP}^{\Sigma_{i-1}^{\mathsf{P}}}$.

Definition 27. *The* polynomial hierarchy*, called* PH*, is defined to be:*

$$\mathsf{PH} = \bigcup_{k=1}^{\infty} \Sigma_k^{\mathsf{P}}$$

$$\begin{tikzcd}
 & \vdots & \\
\mathsf{NP}^{\mathsf{NP}^{\mathsf{NP}}} = \Sigma_3^{\mathsf{P}} & & \Pi_3^{\mathsf{P}} = \mathsf{coNP}^{\mathsf{NP}^{\mathsf{NP}}} \\
 & \mathsf{P}^{\mathsf{NP}^{\mathsf{NP}}} = \Delta_3^{\mathsf{P}} \arrow[ul] \arrow[ur] & \\
\mathsf{NP}^{\mathsf{NP}} = \Sigma_2^{\mathsf{P}} \arrow[uu] \arrow[ur] & & \Pi_2^{\mathsf{P}} = \mathsf{coNP}^{\mathsf{NP}} \arrow[uu] \arrow[ul] \\
 & \mathsf{P}^{\mathsf{NP}} = \Delta_2^{\mathsf{P}} \arrow[ul] \arrow[ur] & \\
\mathsf{NP} = \Sigma_1^{\mathsf{P}} \arrow[uu] \arrow[ur] & & \Pi_1^{\mathsf{P}} = \mathsf{coNP} \arrow[uu] \arrow[ul] \\
 & \Delta_0^{\mathsf{P}} = \Sigma_0^{\mathsf{P}} = \mathsf{P} = \Pi_0^{\mathsf{P}} = \Delta_1^{\mathsf{P}} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Fig. 1. Polynomial Hierarchy, taken from http://commons.wikimedia.org/wiki/File:Polynomial_time_hierarchy.svg. Each arrow represents inclusion: for example, $\Sigma_1^{\mathsf{P}} \subseteq \Delta_2^{\mathsf{P}} \subseteq \Sigma_2^{\mathsf{P}}$.

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.

[MS72] A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th Annual Symposium on Switching and Automata Theory (Swat 1972)*, SWAT '72, pages 125–129, Washington, DC, USA, 1972. IEEE Computer Society.

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