

Chapter 0 - Introduction (Skipped)

Chapter 1 - Regular Languages

Theorem 1.39	Every NFA has an equivalent DFA	pg. 55
Theorem 1.45 - 1.49	Regular languages closure properties	pg. 59
Theorem 1.54	A language is regular iff some regular expression describes it	pg. 66
Example 1.73 - 1.77	Non-regular languages examples	pg. 80-82

- $\{0^n 1^n \mid n \geq 0\}$
- $\{w \mid w \text{ has an equal number of 0s and 1s}\}$
- $\{ww \mid w \in \{0, 1\}^*\}$
- $\{1^{n^2} \mid n \geq 0\}$
- $\{0^i 1^j \mid i > j\}$

NOTE More examples are given under EXERCISES and PROBLEMS, pg. 83 - 93 in the book.

Chapter 2 - Context-Free Languages

Definition 2.8	Chomsky normal form of context-free grammars	pg. 109
Example 2.14 - 2.18	Context-free languages examples	pg. 114-116

- $\{0^n 1^n \mid n \geq 0\}$
- $\{a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i = j \text{ or } i = k\}$
- $\{ww^r \mid w \in \{0, 1\}^*\}$

Theorem 2.20	A language is context free iff some PDA recognizes it	pg. 117
Example 2.36 - 2.38	Non-context-free languages examples (diagrams)	pg. 128-129

- $\{a^n b^n c^n \mid n \geq 0\}$
- $\{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$
- $\{ww \mid w \in \{0, 1\}^*\}$

NOTE More examples are given under EXERCISES and PROBLEMS, pg. 154 - 159.

Chapter 3 - The Church-Turing Thesis

Definition 3.3 - 3.6	Turing machine , recognizable and decidable definitions	pg. 168
Example 3.7 - 3.12	Turing machines examples (with diagrams)	pg. 171-175

- $\{0^{2^n} \mid n \geq 0\}$
- $\{w\#w \mid w \in \{0, 1\}^*\}$
- $\{a^i b^j c^k \mid i \times j = k \text{ and } i, j, k \geq 1\}$
- $\{\#x_1\#x_2\#\dots\#x_l \mid \text{each } x_i \in \{0, 1\}^* \text{ and } x_i \neq x_j \text{ for each } i \neq j\}$
- $\{w \mid w \text{ contains an equal number of 0s and 1s}\}$ (*proof on pg. 191*)

Theorem 3.13 - 3.19	Turing machines variants equivalence	pg. 176-180
• Multitape	\equiv single-tape	
• Nondeterministic	\equiv deterministic	

Figure 3.20	Description of an enumerator	pg. 180
-------------	-------------------------------------------	---------

Theorem 3.21	A language is Turing-recognizable iff it's enumerated	pg. 181
--------------	-------------------------------------------------------------	---------

Hilbert's problem	$\{p \mid p \text{ is a polynomial with an integral root}\}$ is recognizable, but not decidable	pg. 184
--------------------------	-------------------------------------------------------------------------------------------------------	---------

Example 3.23 $\{\langle G \rangle \mid G \text{ is a connected undirected graph} \}$ is decidablepg. 185

EXERCISES Useful theorems, unproved unless otherwise notedpg. 188-190

Note: items with an A symbol instead of a • symbol have a proof on page 191

- 2-PDAs are more powerful than 1-PDAs
- 3-PDAs are NOT more powerful than 2-PDAs
- A Variant: write-once Turing machine
- Variant: Turing machine with doubly infinite tape
- Variant: Turing machine with left reset
- Variant: Turing machine with stay put instead of left
- A language can be recognized by a deterministic queue automaton iff the language is Turing-recognizable
- A Decidable languages closure properties
- A Recognizable languages closure properties
- *NOTE Turing-recognizable languages are NOT closed under complement! Very useful.*

STUDY GUIDE's theorems. *Note: page numbers refer to the study guide, not to the book.*

Example Enumerator of $\{0^k 1^k \mid k \geq 0\}$ pg. 17

Exercise 1.9a Every finite language is decidablepg. 19

Chapter 4 - Decidability

Theorem 4.1 - 4.23 Common languages & decidabilitiespg. 194-210

- | | |
|---------------------------|--------------------------------------------------|
| • A_{DFA} is decidable | • E_{CFG} is decidable |
| • A_{NFA} is decidable | • Every context-free language is decidable |
| • A_{REG} is decidable | • E_{QCFG} is NOT decidable |
| • EQ_{DFA} is decidable | • A_{TM} is NOT decidable |
| • A_{CFG} is decidable | • $\overline{A_{TM}}$ is NOT Turing-recognizable |

Figure 4.10 **regular** \subset **context-free** \subset **decidable** \subset **Turing-recognizable**pg. 201

The diagonalization methodpg. 202

Theorem 4.22 A language is decidable iff it is recognizable and co-recognizablepg. 209

EXERCISES All of the languages below are **decidable**pg. 211-214

Note: items with an A symbol instead of a • symbol have a proof on page 214.

- ALL_{DFA}
- $\{\langle G \rangle \mid G \text{ is a CFG that generates } \varepsilon\}$
- $INFINITE_{DFA} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) \text{ is an infinite language}\}$
- $INFINITE_{PDA} = \{\langle M \rangle \mid M \text{ is a PDA and } L(M) \text{ is an infinite language}\}$
- A $\{\langle M \rangle \mid M \text{ is a DFA that doesn't accept any string containing an odd number of 1s}\}$
- $\{\langle R, S \rangle \mid R \text{ and } S \text{ are regular expressions and } L(R) \subseteq L(S)\}$
- A $\{\langle G \rangle \mid G \text{ is a CFG over } \{0, 1\} \text{ and } 1^* \cap L(G) \neq \emptyset\}$

- $\{\langle G \rangle \mid G \text{ is a CFG over } \{0, 1\} \text{ and } 1^* \subseteq L(G)\}$
 - $\{\langle M \rangle \mid M \text{ is a DFA that accepts } w^R \text{ whenever it accepts } w\}$
- A Say that an NFA is **ambiguous** if it accepts some string along two different computation branches. $AMBIG_{NFA} = \{\langle N \rangle \mid N \text{ is an ambiguous NFA}\}$ is decidable
- A $BAL_{DFA} = \{\langle M \rangle \mid M \text{ is a DFA that accepts strings with an equal number of 0s and 1s}\}$
- $PAL_{DFA} = \{\langle M \rangle \mid M \text{ is a DFA that accepts palindromes}\}$
 - $\{\langle M \rangle \mid M \text{ is a DFA that accepts strings with more 1s than 0s}\}$
 - $\{\langle G, x \rangle \mid G \text{ is a CFG and } x \text{ is a substring of some } y \in L(G)\}$
 - $\{\langle G, k \rangle \mid G \text{ is a CFG and } L(G) \text{ contains exactly } k \text{ strings where } k \geq 0 \text{ or } k = \infty\}$

Problem 4.5 $\overline{E_{TM}}$ is recognizable (*proof on pg. 213*) pg. 211

Problem 4.18 Let C a language. C is Turing-recognizable iff a decidable language D exists such that $C = \{x \mid \exists y (\langle x, y \rangle \in D)\}$ pg. 212

Problem 4.19 The class of decidable languages is **NOT closed** under homomorphism pg. 212

STUDY GUIDE's theorems. *Note: page numbers refer to the study guide, not to the book.*

Exercise 2.6 If G is a CFG in **Chomsky's normal form**, and $w \in L(G)$ is a word of length $n > 0$, then every derivation of w has exactly $2n - 1$ steps pg. 28

Claim L and \overline{L} are both Turing-recognizable iff L is decidable pg. 32

Chapter 5 - Reducibility

Theorem 5.1 - 5.14 Common languages & decidabilities pg. 216-226

- $HALT_{TM}$ is undecidable
- E_{TM} is undecidable
- $REGULAR_{TM}$ is undecidable
- EQ_{TM} is undecidable
- A_{LBA} is decidable
- E_{LBA} is undecidable
- ALL_{CFG} is undecidable

Definition 5.5 Computation history pg. 221

Definition 5.6 Linear bounded automaton (LBA) pg. 221

Lemma 5.8 There are exactly qng^n distinct configurations of an LBA with q states, g symbols and tape of length n pg. 221

NOTE Gray theorems are not in the syllabus

Theorem 5.15 PCP is undecidable pg. 228

Definition 5.17 Computable function pg. 234

Definition 5.20 **Mapping reduction** pg. 235

Theorem 5.22 If $A \leq_m B$ and B is decidable, then A is decidable pg. 236

Corollary 5.23 If $A \leq_m B$ and A is undecidable, then B is undecidable pg. 236

Example 5.24 Reduction from A_{TM} to $HALT_{TM}$ pg. 236

Example 5.25 Reduction from A_{TM} to PCP pg. 237

Example 5.28 If $A \leq_m B$ and B is recognizable, then A is recognizable pg. 237

Corollary 5.29 If $A \leq_m B$ and A is unrecognizable, then B is unrecognizable pg. 238

Theorem 5.30 EQ_{TM} is neither recognizable nor co-recognizable pg. 238

Exercise 5.28 **Rice's theorem** (proof on pg. 243) pg. 241

EXERCISES Useful theorems, unproved unless otherwise noted pg. 239-244

Note: items with an A symbol instead of a • symbol have a proof on page 242-244.

- EQ_{CFG} is undecidable
- EQ_{CFG} is co-Turing-recognizable
- A A_{TM} is NOT mapping reducible to E_{TM}
- A \leq_m is a transitive relation
- A If A is recognizable and $A \leq_m \overline{A}$, then A is decidable
- $\{\langle M \rangle \mid M \text{ is a TM that accepts } w^R \text{ whenever it accepts } w\}$ is undecidable
- The following three languages are undecidable:
 - $\{\langle M, w \rangle \mid M \text{ is a two-tape TM that writes a nonblank symbol on its second tape when it is run on } w\}$
 - $\{\langle M \rangle \mid M \text{ is a two-tape TM that writes a nonblank symbol on its second tape when it is run on some input}\}$
 - $\{\langle M \rangle \mid M \text{ is a single-tape TM that writes a blank symbol over a nonblank symbol when it is run on some input}\}$
- $\{\langle M \rangle \mid M \text{ has a useless state}\}$ is undecidable
- $\{\langle M, w \rangle \mid M \text{ attempts to move its head left when it's on the left-most tape cell}\}$ is undecidable
- $\{\langle M, w \rangle \mid M \text{ attempts to move its head left at any point when run on } w\}$ is decidable
- BB (busy beaver function) is not a computable function
- PCP is decidable over the unary alphabet $\Sigma = \{1\}$
- PCP is undecidable over the binary alphabet $\Sigma = \{0, 1\}$
- SPCP (silly PCP) is decidable
- $AMBIG_{CFG}$ is undecidable (hint: reduction from PCP)
- There exists an undecidable subset of $\{1\}^*$
- A is Turing-recognizable iff $A \leq_m A_{TM}$
- A is decidable iff $A \leq_m 0^*1^*$
- Let $J = \{w \mid \text{either } w = 0x \text{ for some } x \in A_{TM}, \text{ or } w = 1y \text{ for some } y \in \overline{A_{TM}}\}$. Neither J nor \overline{J} is Turing-recognizable
- There exists an undecidable language B where $B \leq_m \overline{B}$
- A_{2DFA} (two-headed finite automaton) is decidable
- E_{2DFA} is undecidable
- $A_{2DIM-DFA}$ (two-dimensional finite automaton) is undecidable
- The following three languages are undecidable (prove using Rice's theorem):
 - A $INFINITE_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is an infinite language}\}$
 - $\{\langle M \rangle \mid M \text{ is a TM and } 1011 \in L(M)\}$
 - $ALL_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \Sigma^*\}$
 - $OVERLAP_{CFG} = \{\langle G, H \rangle \mid G \text{ and } H \text{ are CFGs where } L(G) \cap L(H) \neq \phi\}$
 - $PREFIX-FREE_{CFG} = \{\langle G \rangle \mid G \text{ is a CFG where } L(G) \text{ is a prefix-free}\}$
 - $NECESSARY_{CFG} = \{\langle G, A \rangle \mid A \text{ is a necessary variable in } G\}$ is recognizable and undecidable
 - $MIN_{CFG} = \{\langle G \rangle \mid G \text{ is a minimal CFG}\}$ is recognizable and undecidable

STUDY GUIDE's theorems. *Note: page numbers refer to the study guide, not to the book.*

Exercise 3.1	Proof of Rice's theorem	pg. 41
Exercise 3.2	Each of the two conditions in Rice's theorem is mandatory	pg. 41
Exercise 3.5	In the proof of Theorem 5.9, $(q - 2) \cdot n \cdot g^n + 1$ steps would suffice	pg. 42
Exercise 3.8	Describe a 2DFA that recognizes $\{a^n b^n c^n \mid n \geq 0\}$	pg. 45
Exercise 3.15	Natural number's multiplication is a computable function	pg. 48

Chapter 6 - Advanced Topics in Computability Theory (Skipped)

Chapter 7 - Time Complexity

Definition 7.1	Time complexity	pg. 276
Theorem 7.8	Every $t(n)$ time multitape Turing-machine has an equivalent $O(t^2(n))$ time single-tape Turing-machine, where $t(n) \geq n$	pg. 282
Definition 7.9	Nondeterministic running time	pg. 273
Theorem 7.11	Every $t(n)$ time nondeterministic single-tape Turing-machine has an equivalent $2^{O(t(n))}$ time deterministic single-tape Turing-machine, where $t(n) \geq n$	pg. 284
Definition 7.12	P class	pg. 286
Definition 7.18	Polynomial time verifier	pg. 293
Definition 7.19	NP class	pg. 294
Theorem 7.14 - 7.25	Common problems and their classes	pg. 288-297
<ul style="list-style-type: none"> • $PATH \in P$ • $RELPRIME \in P$ • Every context-free language is in P • $CLIQUE \in NP$ • $SUBSET-SUM \in NP$ 		
Theorem 7.27	$SAT \in P$ iff $P = NP$	pg. 300
Definition 7.28	Polynomial time computable function	pg. 300
Definition 7.29	Polynomial time reduction	pg. 300
Theorem 7.31	If $A \leq_P B$ and $B \in P$, then $A \in P$	pg. 301
Theorem 7.32	3SAT is polynomial time reducible to $CLIQUE$	pg. 302
Definition 7.34	NP-complete	pg. 304
Exercise 7.34	NP-hard definition	pg. 326
Theorem 7.35	If B is NP-complete and $B \in P$, then $P = NP$	pg. 304
Theorem 7.36	If B is NP-complete and $B \leq_P C$ for $C \in NP$, then C is NP-complete ..	pg. 304
Theorem 7.37	SAT is NP-complete	pg. 304
Theorem 7.43 - 7.56	Common NP-complete problems	pg. 311-320
<ul style="list-style-type: none"> • $CLIQUE$ • $VERTEX-COVER$ • $HAMPATH$ • $UHAMPATH$ • $SUBSET-SUM$ 		
EXERCISES	Useful theorems, unproved unless otherwise noted	pg. 322-330

Note: items with an A symbol instead of a • symbol have a proof on page 329.

- P and NP closure properties (*NP star proof on pg. 329*)
- $CONNECTED = \{\langle G \rangle \mid G \text{ is a connected undirected graph}\}$ is in P
- $TRIANGLE = \{\langle G \rangle \mid G \text{ contains a 3-clique}\}$ is in P
- $ALL_{DFA} \in P, EQ_{DFA} \in P$
- $ISO = \{\langle G, H \rangle \mid G \text{ and } H \text{ are isomorphic graphs}\}$ is in NP
- $MODEXP = \{\langle a, b, c, p \rangle \mid a, b, c, p \in \mathbb{N}^+ \text{ such that } a^b \equiv c \pmod{p}\}$ is in P
- $PERM-POWER = \{\langle p, q, t \rangle \mid p = q^t \text{ where } p \text{ and } q \text{ are permutations on } \{1, \dots, k\}\}$ is in P
- $UNARY-SSUM \in P$ (*definition on pg. 323 ex. 7.17*)
- If $P = NP$, then every language $A \in P$, except $A = \phi$ and $A = \Sigma^*$, is NP-complete
- $PRIMES = \{m \mid m \text{ is a prime number in binary}\}$ is in NP
- Proving that $PATH$ is not NP-complete would prove $P \neq NP$
- $SPATH \in P$, and $LPATH$ is NP-complete (*definitions on pg. 324 ex. 7.21*)
- $DOUBLE-SAT = \{\langle \phi \rangle \mid \phi \text{ has at least two satisfying assignments}\}$ is NP-complete
- A $HALF-CLIQUE = \{\langle G \rangle \mid G \text{ has a half clique}\}$ is NP-complete
- $CNF_2 \in P$
- CNF_3 is NP-complete
- $MAX-CUT = \{\langle G, k \rangle \mid G \text{ has a cut of size } k \text{ or more}\}$ is NP-complete
- $3COLOR = \{\langle G \rangle \mid G \text{ is colorable with 3 colors}\}$ is NP-complete
- A $SOLITAIRE = \{\langle G \rangle \mid G \text{ is a winnable game configuration}\}$ is NP-complete
- $\{\langle p \rangle \mid p \text{ is a polynomial in several variables having an integral root}\}$ is **NP-hard**
- If $P = NP$, then these problems can be solved in polynomial time:
 - Satisfying a boolean formula
 - Factoring integers
- A Finding a largest clique in an undirected graph (*proof on pg. 330*)
- Minimizing DFAs can be done in polynomial time (*Details on pg. 327 ex. 7.42*)
- $P \neq NP$ implies that NFAs cannot be minimized in polynomial time
- $2SAT = \{\langle \phi \rangle \mid \phi \text{ is a satisfiable 2cnf-formula}\} \in P$
- $TIME(f(n))$ contains only the regular languages when $f(n) = o(n \log n)$
- P is closed under homomorphism iff $P = NP$

STUDY GUIDE's theorems. *Note: page numbers refer to the study guide, not to the book.*

Exercise 4.2 The problem from ex. 2.6 (does a string derived from a CFG) is in Ppg. 64

Example 2 $COMPOSITES \in P$ pg. 67

Question Switching q_{accept} with q_{reject} will NOT make a TM decide the complement .. pg. 69

Common NP-complete languages pg. 78-79

- $INDEPENDENT-SET$
- $SET-COVER$
- $k-COLORING$
- $k-SET-PACKING$

Chapter 8 - Space Complexity

Definition 8.1	Space complexity	pg. 331
Savitch's theorem	$f(n) \geq \log n$ implies $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$	pg. 334
Definition 8.8	PSPACE-complete (hard)	pg. 337
Theorem 8.9 - 8.14	Common PSPACE-complete problems	pg. 339-348
	<ul style="list-style-type: none"> • <i>TQBF</i> • <i>GG</i> (ez wp) • <i>FORMULA-GAME</i> 	
Definition 8.17	Classes L and NL	pg. 349
Example 8.18	$\{0^k 1^k \mid k \geq 0\} \in L$	pg. 349
Example 8.19	$\text{PATH} \in \text{NL}$	pg. 350
Definition 8.20	Configuration of M on w	pg. 350
Definition 8.21	Log space transducer	pg. 352
Definition 8.22	NL-complete	pg. 352
Theorem 8.23	If $A \leq_L B$ and $B \in L$, then $A \in L$	pg. 352
Corollary 8.24	If any NL-complete language is in L, then $L = \text{NL}$	pg. 353
Theorem 8.25	PATH is NL-complete	pg. 353
Corollary 8.26	$\text{NL} \in \text{P}$	pg. 354
Theorem 8.27	$\text{NL} = \text{coNL}$	pg. 355
EXERCISES	Useful theorems, unproved unless otherwise noted	pg. 357-361
	Note: items with an A symbol instead of a • symbol have a proof on page 361.	
	• $\text{SPACE}(f(n))$ is the same whether we define this class using a single-tape TM or double-tape read-only TM when $f(n) \geq n$	
	• PSPACE closure properties	
A	$A_{\text{DFA}} \in L$	
	• Any PSPACE-hard language is also NP-hard	
A	NL closure properties	
	• $\text{EQ}_{\text{REX}} \in \text{PSPACE}$	
	• If every NP-hard language is also PSPACE-hard, then $\text{PSPACE} = \text{NP}$	
	• A_{LBA} is PSPACE-complete	
	• The following languages are in L:	
	- The language of properly nested parentheses and brackets	
	- $\text{MULT} = \{a\#b\#c \mid a, b, c \in \mathbb{N} \text{ and } a \times b = c\}$	
	- $\text{ADD} = \{\langle x, y, z \rangle \mid x, y, z > 0 \text{ and } x + y = z\}$	
	- $\text{PAL-ADD} = \{\langle x, y \rangle \mid x, y > 0 \text{ and } x + y \text{ is a palindrome}\}$	
	- $\{\langle G \rangle \mid G \text{ is an undirected graph that contains a simple cycle}\}$	
	• $\text{BIPARTITE} = \{\langle G \rangle \mid G \text{ is a bipartite graph}\}$ is in NL	
	• The following languages are NL-complete:	
	- $\text{STRONGLY-CONNECTED} = \{\langle G \rangle \mid G \text{ is a strongly connected graph}\}$	
	- $\text{BOTH}_{\text{NFA}} = \{\langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are NFAs and } L(M_1) \cap L(M_2) \neq \emptyset\}$	

- A_{NFA} E_{NFA} 2SAT
- A $\{\langle G \rangle \mid G \text{ is a directed graph that contains a directed cycle}\}$
- There exists an NL-complete context-free language

STUDY GUIDE's theorems. *Note: page numbers refer to the study guide, not to the book.*

Question	Space complexity of a problem can't be greater than its time complexity	pg. 97
Question	Every PSPACE-hard language is also NP-hard	pg. 100
Exercise 5.4	TM with space complexity $f(n) = o(n)$ and time complexity $2^{O(f(n))}$. . .	pg. 104
Exercise 5.8	If $\text{NL} \in \text{P}$, and $A, B \in \text{NL}$, and $B \notin \{\emptyset, \Sigma^*\}$, then $A \leq_P B$	pg. 105
Exercise 5.9	If $A \leq_L B$ using a log space computable function f , then the length of $f(w)$ is at most polynomial in w	pg. 106
Exercise 5.10	\leq_L is transitive	pg. 107

Section 9.1 - Hierarchy Theorems

Definition 9.1	Space constructible functions	pg. 364
Example 9.2	$\log_2 n, n \log_2 n, n^2$ are space constructible (at least $O(\log n)$)	pg. 364
Space hierarchy theorem	For any space constructible function $f : \mathbb{N} \rightarrow \mathbb{N}$, a language A exists that is decidable in $O(f(n))$ space but not in $o(f(n))$ space	pg. 365
Corollary 9.4	For any two functions $f_1, f_2 : \mathbb{N} \rightarrow \mathbb{N}$ where $f_1(n) = o(f_2(n))$ and f_2 is space constructible, $\text{SPACE}(f_1(n)) \subsetneq \text{SPACE}(f_2(n))$	pg. 367
Definition 9.8	Time constructible functions	pg. 368
Example 9.9	$n \log n, n\sqrt{n}, n^2$ and 2^n are time constructible (at least $n \log n$)	pg. 368
Time hierarchy theorem	For any time constructible function $t : \mathbb{N} \rightarrow \mathbb{N}$, a language A exists that is decidable in $O(t(n))$ space but not decidable in $o(t(n)/\log t(n))$	pg. 369
Corollary 9.11	For any two functions $t_1, t_2 : \mathbb{N} \rightarrow \mathbb{N}$ where $t_1(n) = o(t_2(n)/\log t_2(n))$ and t_2 is time constructible, $\text{TIME}(t_1(n)) \subsetneq \text{TIME}(t_2(n))$	pg. 371
Corollary 9.12	For any $1 \leq \epsilon_1 < \epsilon_2 \in \mathbb{R}$, we have $\text{TIME}(n^{\epsilon_1}) \subsetneq \text{TIME}(n^{\epsilon_2})$	pg. 371
Corollary 9.13	$\text{P} \subsetneq \text{EXPTIME}$	pg. 371
Definition 9.14	EXSPACE-complete	pg. 372
Theorem 9.15	$\text{EQ}_{\text{REX}\uparrow}$ is EXSPACE-complete	pg. 372
EXERCISES	Useful theorems, unproved unless otherwise noted	pg. 389-391

Note: items with an A symbol instead of a • symbol have a proof on page 391.

- A $\text{TIME}(2^n) = \text{TIME}(2^{n+1})$
- A $\text{TIME}(2^n) \subsetneq \text{TIME}(2^{2^n})$
- A $\text{NTIME}(n) \subsetneq \text{PSPACE}$
 - If $\text{NEXPTIME} \neq \text{EXPTIME}$, then $\text{P} \neq \text{NP}$ (*proof in the study guide*)
- A The *pad* function
 - $\text{EQ}_{\text{REX}\uparrow}$ is in P

Chapter 10 - Advanced Topics in Complexity Theory

Theorem 10.1	<i>MIN-VERTEX-COVER</i> approximation algorithm	pg. 394
Theorem 10.2	<i>MAX-CUT</i> 2-optimal approximation algorithm)	pg. 395
Definition 10.3	Probabilistic Turing machine	pg. 396
Definition 10.4	BPP class	pg. 397
Lemma 10.5	The error probability ϵ is valid as long as $\epsilon \in [0, \frac{1}{2})$	pg. 397
Theorem 10.6	If p is prime and $a \in \mathbb{Z}_p^+$, then $a^{p-1} \equiv 1 \pmod{p}$	pg. 399
Lemma 10.7	If p is an odd prime number, $\Pr[\text{PRIME accepts } p] = 1$	pg. 401
Lemma 10.8	If p is an odd composite number, $\Pr[\text{PRIME accepts } p] \leq 2^{-k}$	pg. 402
Theorem 10.9	<i>PRIMES</i> \in <i>BPP</i>	pg. 403
Theorem 10.10	RP class	pg. 403
Exercise 10.7	<i>BPP</i> \subseteq PSPACE	pg. 441

STUDY GUIDE's theorems. *Note: page numbers refer to the study guide, not to the book.*

Problem	Traveling Salesman Problem (TSP)	pg. 124
Problem	Minimal Steiner Tree	pg. 124
Corollary	TSP has no approximation algorithm unless $P = NP$	pg. 126
Problem	0/1-Bin Packing	pg. 105
Exercise 7.7	All of the theorems below	pg. 140-141

- *COMPOSITES* \in RP
- *PRIMES* \in coRP
- RP \subseteq BPP
- coRP \subseteq BPP
- RP \subseteq NP

Appendix A - Closure Properties

	$A \cup B$	$A \cap B$	$A \cdot B$	A^*	\overline{A}	homomorphism
Regular	✓	✓	✓	✓	✓	✓
Context-free	✓	✗	✓	✓	✗	✓
Decidable	✓	✓	✓	✓	✓	✗
Recognizable	✓	✓	✓	✓	✗	✓
P	✓	✓	✓	✓	✓	iff P=NP
NP	✓	✓	✓	✓	iff NP=coNP	
PSPACE	✓	✓	✓	✓	✓	
L	✓	✓	✓	iff L=NL		
NL	✓	✓	✓	✓	✓	
BPP	✓	✓			✓	

Appendix B - Classes Hierarchies

Regular \subset Context-free \subset Decidable \subset Turing-recognizable

$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \subseteq EXPSPACE$
 $\quad \quad \quad \parallel \quad \quad \quad \parallel \quad \quad \quad \parallel$
 $\quad \quad \quad coNL \quad \quad \quad NPSPACE \quad \quad \quad NEXPSPACE$

Note: $NL \subsetneq PSPACE$, $P \subsetneq EXPTIME$, $PSPACE \subsetneq EXPSPACE$



