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

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

NOTE More examples are given under EXERCISES *and* PROBLEMS*, pg. 83-93*

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

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

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

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

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

- $\left\{0^{2^n} \mid n \geq 0\right\}$
- $\left\{w\#w \mid w \in \{0,1\}^*\right\}$
- $\left\{a^i b^j c^k \mid i \times j = k \text{ and } i, j, k \geq 1\right\}$
- $\left\{\#x_1\#x_2\# \cdots \#x_l \mid \text{each } x_i \in \{0,1\}^* \text{ and } x_i \neq x_j \text{ for each } i \neq j\right\}$
- $\{w \mid w$ 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	$\{p \mid p$ has an integral root$\}$ is recognizable but not decidable	pg. 184
Example 3.23	$\{\langle G\rangle \mid G$ is a connected undirected graph $\}$ is decidable	pg. 185

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.*

Chapter 4 - Decidability

- A_{DFA} is decidable
- A_{NFA} is decidable
- A_{REX} is decidable
- EQ_{DFA} is decidable
- A_{CFG} is decidable
- E_{CFG} is decidable
- Every context-free language is decidable
- EQ_{CFG} is NOT decidable
- A_{TM} is NOT decidable
- $\overline{A_{\text{TM}}}$ is NOT Turing-recognizable

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

- ALL_{DFA}
- $\{\langle G\rangle \mid G$ is a CFG that generates $\varepsilon\}$
- $INFINITE_{\text{DFA}} = \{\langle A\rangle \mid A$ is a DFA and $L(A)$ is an infinite language$\}$
- $INFINITE_{\text{PDA}} = \{\langle M\rangle \mid M$ is a PDA and $L(M)$ is an infinite language$\}$

A $\{\langle M\rangle \mid M$ is a DFA that doesn't accept any string containing an odd numbers of 1s $\}$
- $\{\langle R, S\rangle \mid R$ and S are regular expressions and $L(R) \subseteq L(S)\}$

A $\{\langle G\rangle \mid G$ is a CFG over $\{0,1\}$ and $1^* \cap L(G) \neq \phi\}$
- $\{\langle G\rangle \mid G$ is a CFG over $\{0,1\}$ and $1^* \subseteq L(G)\}$

- $\{\langle M\rangle \mid M$ is a DFA that accepts w^R whenever it accepts $w\}$

A Say that an NFA is ***ambiguous*** if it accepts some string along two different computation branches. $AMBIG_{\text{NFA}} = \{\langle N\rangle \mid N$ is an ambiguous NFA$\}$ is decidable

A $BAL_{\text{DFA}} = \{\langle M\rangle \mid M$ is a DFA that accepts strings with an equal number of 0s and 1s$\}$

- $PAL_{\text{DFA}} = \{\langle M\rangle \mid M$ is a DFA that accepts palindromes$\}$
- $\{\langle M\rangle \mid M$ is a DFA that accepts strings with more 1s than 0s$\}$
- $\{\langle G, x\rangle \mid G$ is a CFG and x is a substring of some $y \in L(G)\}$
- $\{\langle G, k\rangle \mid G$ is a CFG and L(G) contains exactly k strings where $k \geq 0$ or $k = \infty\}$

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

Chapter 5 - Reducibility

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

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:

- $\{\langle M, w \rangle \mid M$ is a two-tape TM that writes a nonblank symbol on its second tape when it is run on $w\}$
- $\{\langle M \rangle \mid M$ is a two-tape TM that writes a nonblank symbol on its second tape when it is run on some input$\}$
- $\{\langle M \rangle \mid M$ is a single-tape TM that writes a blank symbol over a nonblank symbol when it is run on some input$\}$

- $\{\langle M \rangle \mid M$ has a *useless state*$\}$ is undecidable
- $\{\langle M, w \rangle \mid M$ attempts to move its head left whilst on the left-most tape cell$\}$ is undecidable
- $\{\langle M, w \rangle \mid M$ 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
- $AMBIG_{\text{CFG}}$ is undecidable (hint: reduction from PCP)
- There exists an undecidable subset of $\{1\}^*$
- A is Turing-recognizable iff $A \leq_m A_{\text{TM}}$
- A is decidable iff $A \leq_m 0^*1^*$
- Let $J = \{w \mid \text{either } w = 0x \text{ for some } x \in A_{\text{TM}}, \text{ or } w = 1y \text{ for some } y \in \overline{A_{\text{TM}}}\}$. 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
- $A_{\text{2DIM-DFA}}$ *(two-dimensional finite automaton)* is undecidable
- The following languages are undecidable *(prove using Rice's theorem)*:

A $INFINITE_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is an infinite language}\}$

- $\{\langle M \rangle \mid M \text{ is a TM and } 1011 \in L(M)\}$
- $ALL_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \Sigma^*\}\}$
- $OVERLAP_{\text{CFG}} = \{\langle G, H \rangle \mid G \text{ and } H \text{ are CFGs where } L(G) \cap L(H) \neq \phi\}$
- $PREFIX - FREE_{\text{CFG}} = \{\langle G \rangle \mid G \text{ is a CFG where } L(G) \text{ is a prefix-free}\}$

- $NECESSARY_{\text{CFG}} = \{\langle G, A \rangle \mid A \text{ is a necessary variable in } G\}$ is recognizable & undecidable
- $MIN_{\text{CFG}} = \{\langle G \rangle \mid G \text{ is a minimal CFG}\}$ is recognizable and undecidable

Chapter 6 - Advanced Topics in Computability Theory (Skipped)

Chapter 7 - Time Complexity

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* = $\{\langle G \rangle \mid G$ is a connected undirected graph$\}$ is in P
- *TRIANGLE* = $\{\langle G \rangle \mid G$ contains a 3-clique$\}$ is in P
- $ALL_{\text{DFA}} \in \text{P}$, $EQ_{\text{DFA}} \in \text{P}$
- *ISO* = $\{\langle G, H \rangle \mid G$ and H are isomorphic graphs$\}$ is in NP
- *MODEXP* = $\{\langle a, b, c, p \rangle \mid a, b, c, p \in \mathbb{N}^+$ such that $a^b \equiv c \pmod{p}\}$ is in P
- *PERM-POWER* = $\{\langle p, q, t \rangle \mid p = q^t$ where p and q are permutations on $\{1, \ldots, k\}\}$ is in P
- *UNARY-SSUM* $\in$ P *(definition on pg. 323 ex. 7.17)*
- **If P = NP**, then every language $A \in \text{P}$, except $A = \phi$ and $A = \Sigma^*$, is NP-complete
- *PRIMES* = $\{m \mid m$ 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* = $\{\langle \phi \rangle \mid \phi$ has at least two satisfying assignments$\}$ is NP-complete

A *HALF-CLIQUE* = $\{\langle G \rangle \mid G$ has a half clique$\}$ is NP-complete

- $CNF_2 \in \text{P}$
- CNF_3 is NP-complete
- *MAX-CUT* = $\{\langle G, k \rangle \mid G$ has a cut of size k or more$\}$ is NP-complete
- *3COLOR* = $\{\langle G \rangle \mid G$ is colorable with 3 colors$\}$ is NP-complete

A *SOLITAIRE* = $\{\langle G \rangle \mid G$ is a winnable game configuration$\}$ is NP-complete

- $\{\langle p \rangle \mid p$ 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* = $\{\langle \phi \rangle \mid \phi$ is a satisfiable 2cnf-formula$\} \in \text{P}$
- 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 (does a string derived from a CFG) is in P	pg. 64
Example 2	*COMPOSITES* $\in$ P	pg. 67
Question	Swapping q_{accept} with q_{reject} will NOT make a TM decide the complement language	pg. 69
Theorems	Common NP-complete languages	pg. 78-79

- *INDEPENDENT-SET*
- *SET-COVER*
- *k-COLORING*
- *k-SET-PACKING*

Chapter 8 - Space Complexity

Note: items with an A symbol instead of a • symbol have a proof on page 361.

- 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
- $A_{\text{DFA}} \in \text{L}$
- Any PSPACE-hard language is also NP-hard

A NL closure properties

- $EQ_{\text{REX}} \in \text{PSPACE}$
- If every NP-hard language is also PSPACE-hard, then PSPACE = NP
- A_{LBA} is PSPACE-complete
- The following languages are in L:
  - The language of properly nested parentheses and brackets
  - $MULT = \{a\#b\#c \mid a, b, c \in \mathbb{N} \text{ and } a \times b = c\}$
  - $ADD = \{\langle x, y, z\rangle \mid x, y, z > 0 \text{ and } x + y = z\}$
  - $PAL\text{-}ADD = \{\langle x, y\rangle \mid x, y > 0 \text{ and } x + y \text{ is a palindrome}\}$
  - $\{\langle G\rangle \mid G \text{ is an undirected graph that contains a simple cycle}\}$
- $BIPARTITE = \{\langle G\rangle \mid G \text{ is a bipartite graph}\}$ is in NL
- The following languages are NL-complete:
  - $STRONGLY\text{-}CONNECTED = \{\langle G\rangle \mid G \text{ is a strongly connected graph}\}$
  - $BOTH_{\text{NFA}} = \{\langle M_1, M_2\rangle \mid M_1, M_2 \text{ are NFAs and } L(M_1) \cap L(M_2) \neq \phi\}$
  - 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	The space complexity of a problem cannot be greater than the problem's time complexity	pg. 97
Question	Every PSPACE-hard language is also NP-hard	pg. 100
Exercise 5.4	Example of a TM with space complexity $f(n) = o(n)$ and time complexity $2^{O(f(n))}$	pg. 104
Exercise 5.8	If NL $\in$ P, and $A, B \in$ NL, and $B \notin \{\phi, \Sigma^*\}$, 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 $f : \mathbb{N} \to \mathbb{N}$, 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 $f_1, f_2 : \mathbb{N} \to \mathbb{N}$ where $f_1(n) = o(f_2(n))$ and f_2 is space constructible, SPACE$(f_1(n)) \subsetneq$ 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 $t : \mathbb{N} \to \mathbb{N}$, 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 $t_1, t_2 : \mathbb{N} \to \mathbb{N}$ where $t_1(n) = o(t_2(n)/\log t_2(n))$ and t_2 is time constructible, TIME$(t_1(n)) \subsetneq$ TIME$(t_2(n))$	pg. 371
Corollary 9.12	For any $1 \leq \epsilon_1 < \epsilon_2 \in \mathbb{R}$, we have TIME$(n^{\epsilon_1}) \subsetneq$ TIME(n^{ϵ_2})	pg. 371
Corollary 9.13	P $\subsetneq$ EXPTIME	pg. 371
Definition 9.14	**EXPSPACE-complete**	pg. 372
Theorem 9.15	$EQ_{\mathrm{REX}\uparrow}$ is EXPSPACE-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 TIME(2^n) = TIME(2^{n+1})
- A TIME$(2^n) \subsetneq$ TIME(2^{2n})
- A NTIME$(n) \subsetneq$ PSPACE
- • If NEXPTIME ≠ EXPTIME, then P ≠ NP *(proof in the study guide)*
- A The *pad* function
- • $E_{\mathrm{REX}\uparrow}$ is in P

Chapter 10 - Advanced Topics in Complexity Theory

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

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

Appendix A - Closure Properties

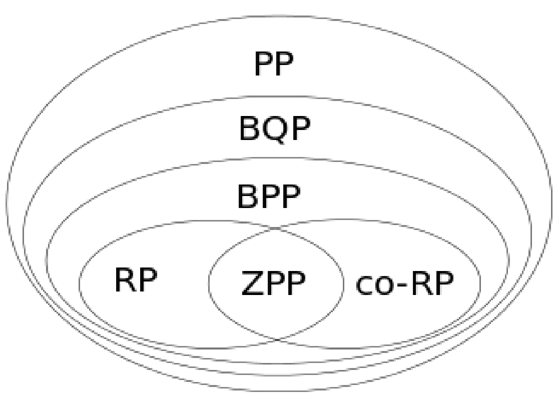
	$A \cup B$	$A \cap B$	$A \cdot B$	A^*	$\overline{A}$	homomorphism
Regular	✓	✓	✓	✓	✓	✓
Context-free	✓	✗	✓	✓	✗	✓
Decidable	✓	✓	✓	✓	✓	✗
Recognizable	✓	✓	✓	✓	✗	✓
P	✓	✓	✓	✓	✓	iff P=NP
NP	✓	✓	✓	✓	iff NP=coNP	
PSPACE	✓	✓	✓	✓	✓	
L	✓	✓	✓	iff L=NL		
NL	✓	✓	✓	✓	✓	
BPP	✓	✓			✓	

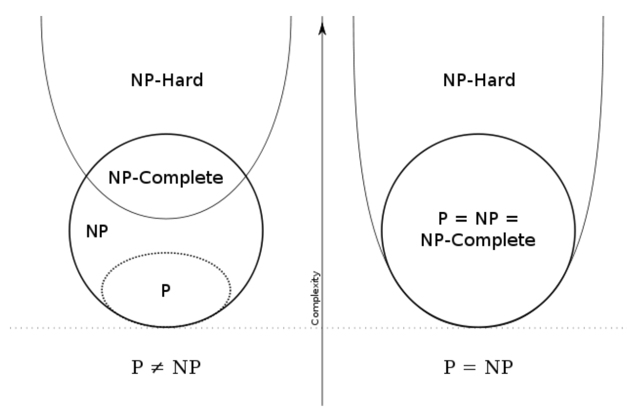
Appendix B - Classes Hierarchies

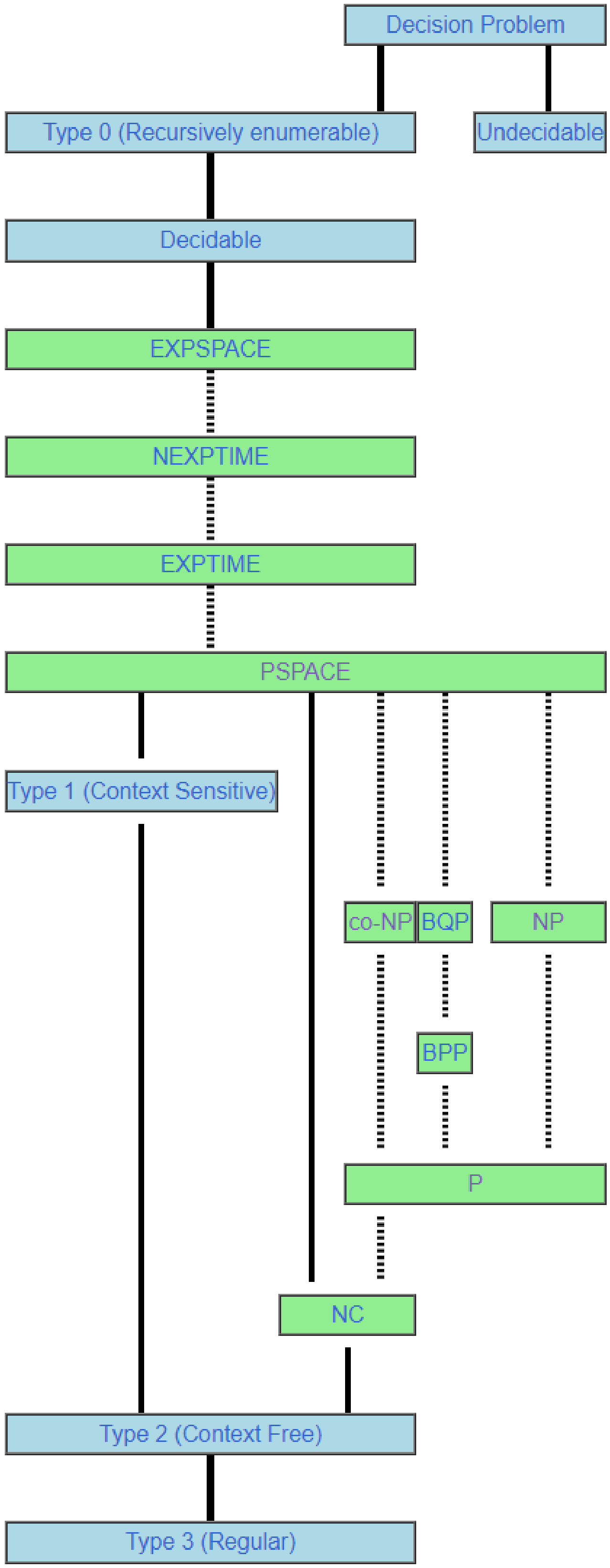
$$\text{Regular} \subset \text{Context-free} \subset \text{Decidable} \subset \text{Turing-recognizable}$$

$$\mathrm{L} \subseteq \underset{\underset{\text{coNL}}{\shortparallel}}{\mathrm{NL}} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \underset{\underset{\text{NPSPACE}}{\shortparallel}}{\mathrm{PSPACE}} \subseteq \mathrm{EXPTIME} \subseteq \underset{\underset{\text{NEXPSPACE}}{\shortparallel}}{\mathrm{EXPSPACE}}$$

Note: $\mathrm{NL} \subsetneq \mathrm{PSPACE}$, $\mathrm{P} \subsetneq \mathrm{EXPTIME}$, $\mathrm{PSPACE} \subsetneq \mathrm{EXPSPACE}$






Decision Problem
Type 0 (Recursively enumerable)
Undecidable
Decidable
EXPSPACE
NEXPTIME
EXPTIME
PSPACE
Type 1 (Context Sensitive)
co-NP
BQP
NP
BPP
P
NC
Type 2 (Context Free)
Type 3 (Regular)