Practice Problems for Final Exam
CS 341: Foundations of Computer Science II
Prof. Marvin K. Nakayama

1. Short answers:

   (a) Define the following terms and concepts:

      i. Union, intersection, set concatenation, Kleene-star, set subtraction, complement
      ii. A set S is closed under an operation f
      iii. Regular language
      iv. Kleene's theorem
      v. Context-free language
      vi. Chomsky normal form
      vii. Church-Turing Thesis
      viii. Turing-decidable language
      ix. Turing-recognizable language
      x. co-Turing-recognizable language
      xi. Countable and uncountable sets
      xii. Language A is mapping reducible to language B, $A \leq_{\mathrm{m}} B$
      xiii. Function $f(n)$ is $O(g(n))$
      xiv. Classes P and NP
      xv. Language A is polynomial-time mapping reducible to language B, $A \leq_{\mathrm{P}} B$.
      xvi. NP-complete
      xvii. NP-hard

   (b) Give the transition functions δ (i.e., specify the domains and ranges) of a DFA, NFA, PDA, Turing machine and nondeterministic Turing machine.

   (c) Explain the "P vs. NP" problem.

2. Recall that $A_{\mathrm{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM that accepts string } w \}$.

   (a) Prove that A_{TM} is undecidable. You may not cite any theorems or corollaries in your proof.

   (b) Show that A_{TM} is Turing-recognizable.

3. Each of the languages below in parts (a), (b), (c), (d) is of one of the following types:

   Type REG. It is regular.
   Type CFL. It is context-free, but not regular.
   Type DEC. It is Turing-decidable, but not context-free.

   For each of the following languages, specify which type it is. Also, follow these instructions:

   - If a language L is of Type REG, give a regular expression **and** a DFA (5-tuple) for L.
   - If a language L is of Type CFL, give a context-free grammar (4-tuple) **and** a PDA (6-tuple) for L. **Also, prove that L is not regular.**
   - If a language L is of Type DEC, give a description of a Turing machine that decides L. **Also, prove that L is not context-free.**

   (a) $A = \{ w \in \Sigma^* \mid w = \text{reverse}(w) \text{ and the length of } w \text{ is divisible by } 4 \}$, where $\Sigma = \{0, 1\}$.

   Circle one type: REG CFL DEC

(b) $B = \{ b^n a^n b^n \mid n \geq 0 \}$.

Circle one type: REG CFL DEC

(c) $C = \{ w \in \Sigma^* \mid n_a(w) \bmod 4 = 1 \}$, where $\Sigma = \{a, b\}$ and $n_a(w)$ is the number of a's in string w. For example, $n_a(babaabb) = 3$. Also, recall $j \bmod k$ returns the remainder after dividing j by k, e.g., $3 \bmod 4 = 3$, and $9 \bmod 4 = 1$.

Circle one type: REG CFL DEC

(d) $D = \{ b^n a^n b^k c^k \mid n \geq 0, \, k \geq 0 \}$. [Hint: Recall that the class of context-free languages is closed under concatenation.]

Circle one type: REG CFL DEC

4. Each of the languages below in parts (a), (b), (c), (d) is of one of the following types:

Type DEC. It is Turing-decidable.
Type TMR. It is Turing-recognizable, but not decidable.
Type NTR. It is not Turing-recognizable.

For each of the following languages, specify which type it is. Also, follow these instructions:

- If a language L is of Type DEC, give a description of a Turing machine that decides L.
- If a language L is of Type TMR, give a description of a Turing machine that recognizes L. **Also, prove that L is not decidable.**
- If a language L is of Type NTR, give a proof that it is not Turing-recognizable.

In each part below, if you need to prove that the given language L is decidable, undecidable, or not Turing-recognizable, you must give an explicit proof of this; i.e., do not just cite a theorem that establishes this without a proof. However, if in your proof you need to show another language L' has a particular property and there is a theorem that establishes this, then you may simply cite the theorem for L' without proof.

(a) $EQ_{\text{TM}} = \{ \langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are TMs with } L(M_1) = L(M_2) \}$. [Hint: show $\overline{A_{\text{TM}}} \leq_{\text{m}} EQ_{\text{TM}}$.]

Circle one type: DEC TMR NTR

(b) $HALT_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM that halts on input } w \}$. [Hint: modify the universal TM to show $HALT_{\text{TM}}$ is Turing-recognizable.]

Circle one type: DEC TMR NTR

(c) $EQ_{\text{DFA}} = \{ \langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are DFAs with } L(M_1) = L(M_2) \}$.

Circle one type: DEC TMR NTR

(d) $\overline{A_{\text{TM}}}$, where $A_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM that accepts string } w \}$.

Circle one type: DEC TMR NTR

5. Let $L_1, L_2, L_3, \ldots$ be an infinite sequence of regular languages, each of which is defined over a common input alphabet Σ. Let $L = \cup_{k=1}^{\infty} L_k$ be the infinite union of $L_1, L_2, L_3, \ldots$. Is it always the case that L is a regular language? If your answer is YES, give a proof. If your answer is NO, give a counterexample. Explain your answer. [Hint: Consider, for each $k \geq 0$, the language $L_k = \{a^k b^k\}$.]

6. Let L_1, L_2, and L_3 be languages defined over the alphabet $\Sigma = \{a, b\}$, where

   - L_1 consists of all possible strings over Σ except the strings $w_1, w_2, \ldots, w_{100}$; i.e., start with all possible strings over the alphabet, take out 100 particular strings, and the remaining strings form the language L_1;
   - L_2 is recognized by an NFA; and
   - L_3 is recognized by a PDA.

   Prove that $(L_1 \cap L_2)L_3$ is a context-free language. [Hint: First show that L_1 and L_2 are regular. Also, consider $\overline{L_1}$, the complement of L_1.]

7. Write Y or N in the entries of the table below to indicate which classes of languages are closed under which operations.

Operation	Regular languages	CFLs	Decidable languages	Turing-recognizable languages
Union				
Intersection				
Complementation				

8. Consider the following context-free grammar G in Chomsky normal form:

$$\begin{aligned} S &\rightarrow a \mid YZ \\ Z &\rightarrow ZY \mid a \\ Y &\rightarrow b \mid ZZ \mid YY \end{aligned}$$

   Use the CYK (dynamic programming) algorithm to fill in the following table to determine if G generates the string $babba$. Does G generate $babba$?

9. Recall that

$$\begin{aligned} \mathit{CLIQUE} &= \{ \langle G, k \rangle \mid G \text{ is an undirected graph with a } k\text{-clique} \}, \\ \mathit{3SAT} &= \{ \langle \phi \rangle \mid \phi \text{ is a satisfiable 3cnf-function} \}. \end{aligned}$$

   Show that *CLIQUE* is NP-Complete by showing that $\mathit{CLIQUE} \in \mathrm{NP}$ and $\mathit{3SAT} \leq_{\mathrm{P}} \mathit{CLIQUE}$. Explain your reduction for the general case and not just for a specific example. Be sure to prove your reduction works and that it requires polynomial time. Also, be sure to provide proofs of these results, and don't just cite a theorem.

10. Recall that

$$\mathit{ILP} = \{ \langle A, b \rangle \mid \text{matrix } A \text{ and vector } b \text{ satisfy } Ay \leq b \text{ with } y \text{ and integer vector} \}.$$

   Show that *ILP* is NP-Complete by showing that $\mathit{ILP} \in \mathrm{NP}$ and $\mathit{3SAT} \leq_{\mathrm{P}} \mathit{ILP}$. Explain your reduction for the general case and not just for a specific example. Be sure to prove your reduction works and that it requires polynomial time. Also, be sure to provide proofs of these results, and don't just cite a theorem.

List of Theorems

Thm 1.A. The class of regular languages is closed under union.

Thm 1.B. The class of regular languages is closed under concatenation.

Thm 1.C. Every NFA has an equivalent DFA.

Thm 1.D. The class of regular languages is closed under Kleene-star.

Thm 1.E. (Kleene's Theorem) Language A is regular iff A has a regular expression.

Thm 1.F. If A is finite language, then A is regular.

Thm 1.G. The class of regular languages is closed under intersection.

Thm 1.H. The class of regular languages is closed under complementation.

Thm 1.I. (Pumping lemma for regular languages) If A is regular language, then $\exists$ number p where, if $s \in A$ with $|s| \geq p$, then can split $s = xyz$ satisfying the conditions (1) $xy^iz \in A$ for each $i \geq 0$, (2) $|y| > 0$, and (3) $|xy| \leq p$.

Thm 2.A. Every CFL can be described by a CFG $G = (V, \Sigma, R, S)$ in Chomsky normal form, i.e., each rule in G has one of two forms: $A \to BC$ or $A \to x$, where $A \in V$, $B, C \in V - \{S\}$, $x \in \Sigma$, and we also allow the rule $S \to \varepsilon$.

Thm 2.B. If A is a regular language, then A is also a CFL.

Thm 2.C. A language is context free iff some PDA recognizes it.

Thm 2.D. (Pumping lemma for CFLs) For every CFL L, $\exists$ pumping length p such that $\forall$ strings $s \in L$ with $|s| \geq p$, can split $s = uvxyz$ with (1) $uv^ixy^iz \in L \ \forall \ i \geq 0$, (2) $|vy| \geq 1$, (3) $|vxy| \leq p$.

Thm 2.E. The class of CFLs is closed under union.

Thm 2.F. The class of CFLs is closed under concatenation.

Thm 2.G. The class of CFLs is closed under Kleene-star.

Thm 3.A. For every multi-tape TM M, there is a single-tape TM M' such that $L(M) = L(M')$.

Thm 3.B. Every NTM has an equivalent deterministic TM.

Cor 3.C. Language L is Turing-recognizable iff an NTM recognizes it.

Thm 3.D. A language is enumerable iff some enumerator enumerates it.

Church-Turing Thesis. Informal notion of algorithm corresponds to a Turing machine that always halts.

Thm 4.A. $A_{\text{DFA}} = \{ \langle B, w \rangle \mid B \text{ is a DFA that accepts string } w \}$ is Turing-decidable.

Thm 4.B. $A_{\text{NFA}} = \{ \langle B, w \rangle \mid B \text{ is an NFA that accepts string } w \}$ is Turing-decidable.

Thm 4.C. $A_{\text{REX}} = \{ \langle R, w \rangle \mid R \text{ is a regular expression that generates string } w \}$ is Turing-decidable.

Thm 4.D. $E_{\text{DFA}} = \{ \langle B \rangle \mid B \text{ is a DFA with } L(B) = \emptyset \}$ is Turing-decidable.

Thm 4.E. $EQ_{\text{DFA}} = \{ \langle A, B \rangle \mid A \text{ and } B \text{ are DFAs with } L(A) = L(B) \}$ is Turing-decidable.

Thm 4.F. $A_{\text{CFG}} = \{ \langle G, w \rangle \mid G \text{ is a CFG that generates string } w \}$ is Turing-decidable.

Thm 4.G. $E_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG with } L(G) = \emptyset \}$ is Turing-decidable.

Thm 4.H. Every CFL is Turing-decidable.

Thm 4.I. $A_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM that accepts string } w \}$ is undecidable.

Thm 4.J. The set $\mathcal{R}$ of all real numbers is uncountable.

Cor 4.K. Some languages are not Turing-recognizable.

Thm 4.L. A language is decidable iff it is both Turing-recognizable and co-Turing-recognizable.

Cor 4.M. $\overline{A_{\mathrm{TM}}}$ is not Turing-recognizable.

Thm 5.A. $\mathit{HALT}_{\mathrm{TM}} = \{ \langle M, w\rangle \mid M$ is a TM that halts on $w \}$ is undecidable.

Thm 5.B. $E_{\mathrm{TM}} = \{ \langle M\rangle \mid M$ is a TM with $L(M) = \emptyset \}$ is undecidable.

Thm 5.C. $\mathit{REG}_{\mathrm{TM}} = \{ \langle M\rangle \mid M$ is a TM and $L(M)$ is regular$\}$ is undecidable.

Thm 5.D. $\mathit{EQ}_{\mathrm{TM}} = \{ \langle M_1, M_2\rangle \mid M_1$, M_2 are TMs with $L(M_1) = L(M_2) \}$ is undecidable.

Thm 5.E. (Rice's Thm.) Let $\mathcal{P}$ be any subset of the class of Turing-recognizable languages such that $\mathcal{P} \neq \emptyset$ and $\overline{\mathcal{P}} \neq \emptyset$. Then $L_{\mathcal{P}} = \{ \langle M\rangle \mid L(M) \in \mathcal{P} \}$ is undecidable.

Thm 5.F. If $A \leq_{\mathrm{m}} B$ and B is Turing-decidable, then A is Turing-decidable.

Cor 5.G. If $A \leq_{\mathrm{m}} B$ and A is undecidable, then B is undecidable.

Thm 5.H. If $A \leq_{\mathrm{m}} B$ and B is Turing-recognizable, then A is Turing-recognizable.

Cor 5.I. If $A \leq_{\mathrm{m}} B$ and A is not Turing-recognizable, then B is not Turing-recognizable.

Thm 5.J. $E_{\mathrm{TM}} = \{ \langle M\rangle \mid M$ is a TM with $L(M) = \emptyset \}$ is not Turing-recognizable.

Thm 5.K. $\mathit{EQ}_{\mathrm{TM}} = \{ \langle M_1, M_2\rangle \mid M_1, M_2$ are TMs with $L(M_1) = L(M_2) \}$ is neither Turing-recognizable nor co-Turing-recognizable.

Thm 7.A. Let $t(n)$ be a function with $t(n) \geq n$. Then any $t(n)$-time multi-tape TM has an equivalent $O(t^2(n))$-time single-tape TM.

Thm 7.B. Let $t(n)$ be a function with $t(n) \geq n$. Then any $t(n)$-time NTM has an equivalent $2^{O(t(n))}$-time deterministic 1-tape TM.

Thm 7.C. $\mathit{PATH} \in$ P.

Thm 7.D. $\mathit{RELPRIME} \in$ P.

Thm 7.E. Every CFL is in P.

Thm 7.F. A language is in NP iff it is decided by some nondeterministic polynomial-time TM.

Cor 7.G. $\mathrm{NP} = \bigcup_{k \geq 0} \mathrm{NTIME}(n^k)$

Thm 7.H. $\mathit{CLIQUE} \in$ NP.

Thm 7.I. $\mathit{SUBSET\text{-}SUM} \in$ NP.

Thm 7.J. If $A \leq_{\mathrm{P}} B$ and $B \in \mathrm{P}$, then $A \in \mathrm{P}$.

Thm 7.K. $\mathit{3SAT}$ is polynomial-time reducible to CLIQUE.

Thm 7.L. If there is an NP-Complete problem B and $B \in \mathrm{P}$, then $\mathrm{P} = \mathrm{NP}$.

Thm 7.M. If B is NP-Complete and $B \leq_{\mathrm{P}} C$ for $C \in \mathrm{NP}$, then C is NP-Complete.

Thm 7.N. (Cook-Levin Thm.) SAT is NP-Complete.

Cor 7.O. $\mathit{3SAT}$ is NP-Complete.

Cor 7.P. CLIQUE is NP-Complete.

Thm 7.Q. ILP is NP-Complete.