17 Practice Questions

Table of contents

Practice question sets

Due 8/12

1. Fully parenthesize propositional formula $P \wedge Q \vee R$. That is, add parentheses so that the structure is determined by the parentheses without the need for rules of precedence.
2. Fully parenthesize propositional formula $P \wedge Q \to R \wedge S$.
3. Fully parenthesize propositional formula $P \to Q \to R$.
4. How many rows are in the truth table of propositional formula $P \to (Q \to (\neg R \to S))$?
5. Construct a truth table of $P \to (Q \wedge P)$.
6. Using a truth table, show that $P \to (Q \to P)$ is a tautology.
7. Using a truth table, show that $(P \wedge Q) \to R \equiv (P \wedge \neg R) \to \neg Q$ is a tautology.
8. Suppose that the domain of discourse is the set of all integers. Say whether each of the following is true or false.
   (a) $\forall n \exists m (n^2 < m)$.
   (b) $\exists m \forall n (n^2 < m)$.
   (c) $\exists n \exists m (n^2 + m^2 = 5)$.
   (d) $\exists n \exists m (n^2 + m^2 = 6)$.
   (e) $\forall n \forall m \exists r (m + n = 2p)$.
9. Are first-order formulas $\exists x P(x) \wedge \exists x Q(x)$ and $\exists x(P(x) \wedge Q(x))$ logically equivalent? (That is, is

$$\exists x P(x) \wedge \exists x Q(x) \equiv \exists x(P(x) \wedge Q(x))$$

valid?) If so, explain why. If not, give definitions of $P(x)$ and $Q(x)$ where they are different.

Due 8/14

1. Suppose that the domain of discourse is the set of integers. Prove that, if $k + n$ is even and $n + m$ is even then $k + m$ is even.

2. Suppose that the domain of discourse is the set of real numbers. Let $R(x)$ be defined to mean "x is a rational number." Prove

$$\exists x \exists y (\neg R(x) \wedge \neg R(y) \wedge R(xy))$$

.

3. Using proof by contradiction, prove that, if n is an integer where $n^3 + 5$ is odd, then n is even.

Due 8/17

1. Write an enumeration of the members of set $\{x \mid x \in \mathcal{Z} \wedge x \geq 0 \wedge x > x^2 - 5\}$.

2. Give an enumeration of the members of each of the following sets.

   (a) $\{1, 3, 5, 6\} \cup \{2, 3, 5, 9\}$

   (b) $\{1, 3, 5, 6\} \cap \{2, 3, 5, 9\}$

   (c) $\{1, 3, 5, 6\} - \{2, 3, 5, 9\}$

3. True or false?

   (a) $\{2, 4, 6\} \subseteq \{2, 4, 6, 8\}$.

   (b) $\{2, 4, 6\} \in \{2, 4, 6, 8\}$.

   (c) $S - S = \{\}$ for every set S.

   (d) $2 \in \{2\}$.

   (e) $\{\} \in \{\}$.

   (f) $\{\} \subseteq \{\}$.

   (g) The empty set is a language.

   (h) The empty string is a language.

   (i) Every language is finite.

   (j) Every alphabet is finite.

   (k) A string can be infinitely long.

   (l) Some languages contain the empty string.

   (m) Every language contains the empty string.

   (n) A nonempty set of languages is not a language.

Due 8/19

1. Draw a transition diagram for a FSM with alphabet $\{a, b\}$ that decides language $\{"aab"\}$.
2. Draw a transition diagram for a FSM with alphabet $\{a, b\}$ that decides the set of all strings that begin with aa.
3. Draw a transition diagram for a FSM with alphabet $\{a, b\}$ that decides the set of all strings that end on aa. Number the states and say what $\text{Set}(q)$ is for each state q.
4. Draw a transition diagram for a FSM with alphabet $\{a, b\}$ that decides the set of all strings that contain exactly three bs. Number the states and say what $\text{Set}(q)$ is for each state q.
5. Draw a transition diagram for a FSM with alphabet $\{a, b, c\}$ that decides the set of all strings that contain $cacab$ as a contiguous substring.

Due 8/21

1. Let $L_1 = \{x \mid x \in \{a, b\}^* \wedge x$ has the same number of a's as b's$\}$. Prove that L_1 is not regular.

Due 8/24

1. Let $L_2 = \{www \mid w \in \{a, b, c\}^*\}$. Prove that L_2 is not regular.

Due 8/26

1. Is $\{\}$ computable? Justify your answer.
2. A positive integer n is *perfect* if n is the sum of its proper divisors. For example, 6 is perfect because $6 = 1 + 2 + 3$. Show that the set of perfect integers is computable.
3. Let $B = \{n \mid n \text{ is a positive integer that can be expressed as the sum of two prime numbers}\}$. For example, $8 \in B$ since $8 = 5 + 3$. Show that B is computable.
4. Prove that every finite language is computable.
5. Give an example of an infinite computable set.

Due 8/28

1. What properties does program I need to have for I to be an interpreter (for the programming language that has been chosen as the standard one for programs)?

2. Consider the following definition of a "semi-computable" language over alphabet Σ. For any program p, define $\text{Acc}(p) = \{x \mid \text{Run}(p, x) \cong 1\}$. Say that language A is semi-computable if there exists a program p where $\text{Acc}(p) = A$.

   Is the definition of a semi-computable language equivalent to the definition of a computable language? That is, is it true that A is computable if and only if A is semi-computable? Justify your answer.

3. Let A be the decision problem:

   Input. Two FSMs M_1 and M_2, both with alphabet Σ.
   Question. Is $L(M_1) \cup L(M_2) = \Sigma^*$?

   Show that A is computable.

4. Let B be the following decision problem:

   Input. A polynomial p of degree 3 with integer coefficients in a single variable, x.
   Question. Does there exist a value of x for which $p = 0$?

   Show that B is computable.

5. Let INFINITE be the following decision problem
   Input. A FSM M.
   Question. Does M accept infinitely many strings?

   Show that INFINITE is computable. It is not necessary to go into details about how to solve clearly solvable problems about graphs.

Due 8/31

1. Are all infinite languages uncomputable? Justify your answer.

2. Suppose that A and B are languages. What is the definition of a Turing reduction from A to B?

3. Define

$$\begin{aligned} L_1 &= \{p \mid \text{Run}(p, p)\downarrow\} \\ L_2 &= \{(p, x) \mid \text{Run}(p, x)\downarrow\} \end{aligned}$$

   Give a Turing reduction from L_1 to L_2.

4. Define

$$\begin{aligned} L_1 &= \{p \mid \text{Run}(p, 1)\downarrow\} \\ L_2 &= \{p \mid \text{Run}(p, 1)\uparrow\} \end{aligned}$$

   Give a Turing reduction from L_1 to L_2.

5. Suppose that A and B are languages over alphabet Σ where B is computable and $A \subseteq B$. Is it necessarily true that A is computable? Justify your answer.

   Think this out. Suppose that $B = \Sigma^*$. What are the subsets of B?

Due 9/2

1. What is the definition of a mapping reduction from language A to language B?
2. What is the definition of $A \leq_p B$?
3. Suppose that A and B are both computable languages over alphabet Σ. Show that $A \leq_t B$.
4. Suppose that A and B are both computable languages over alphabet Σ where $B \neq \{\}$ and $B \neq \Sigma^*$. Show that $A \leq_m B$.
5. Suppose that $A \leq_t B$ and B is a regular language. Does that imply that A is also a regular language? Justify your answer.
6. Suppose that A is a language where $A \leq_t$ HLT. Can you conclude that A is uncomputable?
7. Define

$$\begin{aligned} L_1 &= \{p \mid \text{Run}(p, p)\downarrow\} \\ L_2 &= \{(p, x) \mid \text{Run}(p, x)\downarrow\} \end{aligned}$$

Give a mapping reduction from L_1 to L_2.

Due 9/4

1. Show that language $\{p \mid \text{Run}(p, \texttt{"}aa\texttt{"}) \cong 0 \text{ and } \text{Run}(p, \texttt{"}bb\texttt{"}) \cong 1\}$ is not computable.

2. Let $A = \{p \mid \text{Run}(p, \texttt{"}bbb\texttt{"})\downarrow\}$. Give a mapping reduction from A to HLT.

3. Define $B = \{p \mid L(p) \text{ is a regular language}\}$. Is B computable? Justify your answer.

4. For the purposes of this exercise, assume that the output of a program is an integer. Suppose $A = \{p \mid \text{Run}(p, 0) \cong 5\}$ and $B = \{p \mid \text{Run}(p, 0) = 10\}$. Give a mapping reduction from A to B. Be sure that you know what properties the reduction needs to have before you start to describe the reduction.

Due 9/9

1. What is the definition of class P?

2. Suppose that L is a set of positive integers and suppose that there is an algorithm that takes an integer n and tells you whether $n \in L$ in time $O(n^2)$. Can you conclude that L is in P based on that? Explain why or why not.

3. Suppose that L is a set of strings, and suppose that there is an algorithm that takes a string x and tells you whether $x \in L$ in time $O(2^n)$, where $n = |x|$. Can you conclude that L is not in P based on that? Explain why or why not.

4. A *triangle* in a simple graph consists of three mutually adjacent vertices. Show that the problem of determining whether a simple graph contains a triangle is in P.

Due 9/11

1. What is the definition of NP?

2. Is $\{\}$ in NP?

3. Suppose that Σ is an alphabet. Is Σ^* in NP?

4. A *bijection* is a function that is one-to-one and onto. Two simple graphs $G = (V, E)$ and $H = (W, F)$ are *isomorphic* if $|V| = |W|$ and there is a bijection $f : V \to W$ such that, for every pair of vertices a and b in V, $\{a, b\} \in E \leftrightarrow \{f(a), f(b)\} \in F$.

   The *Graph Isomorphism Problem* (GIP) is the following decision problem.

   > **Input.** Simple graphs G and H.
   > **Question.** Are G and H isomorphic?

   Show that GIP is in NP.

5. Let DOUBLE-SATPL be the following decision problem.

   > **Input.** A propositional formula ϕ.
   > **Question.** Do there exist two different truth-value assignments a and b where $(a \dashv \phi) = \mathrm{T}$ and $(b \dashv \phi) = \mathrm{T}$? That is, can ϕ be made true by two different choices of the values of its variables?

   Show that DOUBLE-SAT is in NP.

Due 9/14

1. What is the definition of a polynomial-time mapping reduction from language A to language B?

2. What is the definition of notation $A \le_p B$?

3. Suppose that $A \in$ P and $A \le_p B$. Can you conclude that $B \in$ P?

4. Suppose that $B \in$ P and $A \le_p B$. Can you conclude that $A \in$ P?

5. Suppose that P = NP. Show that, for every $A \in$ NP, $A \le_p \{1\}$.

6. SATPL is the following decision problem.

   Input. A propositional formula ϕ.
   Question. Does there exist a truth-value assignment a where $(a \dashv \phi) =$ T? That is, is it possible to choose values for the propositional variables in ϕ so that ϕ is true?

   Show that SATPL $\le_p$ DOUBLE-SATPL by giving a polynomial-time mapping reduction from SATPL to DOUBLE-SATPL. (DOUBLE-SATPL is defined above.) (**Hint.** Add an extra variable.)

7. Give an example of a decision problem that is not in NP. Justify your answer.

Due 9/16

1. What is the definition of an NP-complete problem?
2. Does there exist a decision problem that is not in $\mathrm{P} \cup \mathrm{NP}$? Justify your answer.
3. Show that DOUBLE-SATPL $\leq_p$ SAT. You are not required to give a polynomial-time mapping reduction from DOUBLE-SATPL to SATPL. But give an air tight argument that such a mapping reduction must exist.
4. Let A be the set of all natural numbers that are prime. Does there exist a polynomial-time mapping reduction from A to SAT?
5. Suppose B is in NP and A is NP-complete and $A \leq_p B$. Can you conclude that B is NP-complete?
6. Suppose that A is NP-complete and $A \subseteq B$. Can you conclude that B is NP-complete? Justify your answer.

Due 9/18

1. If P = NP, is SAT NP-complete?

2. Is SAT known to be NP-complete, or is SAT only conjectured to be NP-complete?

3. Is it known that SAT $\notin$ P?

4. Show that DOUBLE-SATPL is NP-complete. You can appeal to answers to prior exercises without repeating them.

5. The Hitting Set Problem (HSP) is as follows.

   Input. Positive integers N and K; and a list of sets x_1, $\ldots$, x_m, where $x_i \subseteq \{1, \ldots, N\}$ for $i = 1, \ldots, m$.
   Question. Does there exist a set $S \subseteq \{1, \ldots, N\}$ where $|S| \leq K$ and $x_i \cap S \neq \{\}$ for $i = 1, \ldots, m$. That is, S must contain at least one member of each set x_i.

   (a) Give a polynomial-time evidence checker for HSP. Be sure that it is correct for every input and that your description is clear and easy to understand.

   (b) Give a polynomial-time reduction from the the Vertex Cover Problem (VCP) to HSP. Be sure that the reduction is correct for all possible inputs. Describe the reduction in a clear, readable way. Be sure that I can find your definition of the reduction. Just words describing what the reduction might do are not adequate.

   (c) Are the results of parts (a) and (b) of this problem sufficient for you to conclude that HSP is NP-complete? Explain why or why not.

   (d) Does there exist a polynomial-time reduction from HSP to VCP? Either argue that there probably is no such reduction or explain why there must exist such a reduction.

Due 9/21

1. The Partition Problem (PP) is described in Section 13. Give a polynomial-time reduction from PP to the Subset Sum Problem.

2. See above for the definition of isomorphic graphs.

   ¡p¿Suppose that G and H are simple graphs. Say that H is isomorphic to a subgraph of G provided it is possible to remove zero or more vertices and zero or more edges from G and get a graph that is isomorphic to H. (When you remove a vertex v, you must also remove all edges that are incident on v.)

   The Subgraph Isomorphism Problem (SIP) is the following decision problem.

   > **Input.** Simple graphs G and H.
   > **Question.** Is H isomorphic to a subgraph of G?

   Prove that SIP is NP-complete. (**Hint.** Reduce from the Clique Problem.)

Due 9/23

1. What is the definition of Co-NP?
2. Assume that P $\neq$ NP. Is the Validity Problem for Propositional Logic NP-complete? Explain.
3. Give a polynomial-time reduction from the Hamilton Cycle Problem to the Subgraph Isomorphism Problem (above).
4. Suppose that a particular university chooses a set C of classes to offer in a given term and has N time slots in which to schedule classes. Each student selects a set of classses that he or she wants to take.

   The Class Scheduling Problem (CSP) is the following decision problem.

   > **Input.** Positive integer N, set of classes $C = \{c_1, \ldots, c_m\}$, and list of sets $s_1, \ldots, s_k$ where $s_i \subseteq C$ is the set of classes that student i wants to take.
   > **Question.** Does there exist a way to schedule classes into time slots so that no student wants to take two classes that are assigned to the same time slot?

   (a) Prove that CSP is in NP by giving a polynomial-time evidence checker for CSP.

   (b) Give a polynomial-time reduction from the Graph Coloring Problem (GCP) to CSP. (**Hint.** Think about what corresponds to a vertex, what corresponds to an edge and what corresponds to a color.)