

Practice Final Exam

Note: This is a practice final, based on exams used in previous courses. The practice exam is intended to help prepare you for the final, so we think it is useful to first try to take it yourself under exam-like conditions, and then to go back to the problems you had difficulty with to study them without time limits and with help from others.

The collaboration and resource policy for the final exam will be the same as what it was on the midterm:

For this exam, you must **work alone**. You are not permitted to obtain help from people other than asking clarifying questions of the course staff. You are not permitted to provide help to others taking the exam. **You may not use any resources other than your brain and body, the one page of notes you prepared, and a simple writing implement like a pen or pencil.**

You may use this practice exam however you wish, but we think it will be most useful to first try it under exam-like conditions—prepare your page of notes to use and don't use any other resources during the exam (but you may find it helpful to keep track of things you wish you had included in your notes), and try to solve all the problems yourself within an exam time limit (we would suggest no more than two hours for this practice exam, although you will have three hours for the actual final).

True, False, or Unknown

1. For each of the following, circle one of the choices to indicate whether the statement is known to be *True*, is known to be *False*, or *Unknown* if its validity depends on something that is either currently unknown or not specified in the question.

Then, write a short justification to support your answer. When your answer is *Unknown*, your answer should make it clear what unknown the validity of the statement depends on (for example, that it is equivalent to a statement whose truth is currently unknown to anyone).

(a) The function, $XOR : \{0, 1\}^* \rightarrow \{0, 1\}$, which outputs the parity of all the input bits, is in the complexity class P.

Circle one:

True

False

Unknown

Justification (≤ 5 words):

(b) The function, XOR (from the previous question), is in the complexity class NP.

Circle one:

True

False

Unknown

Justification (3 symbols):

(c) The function, XOR (from the previous question), is in the complexity class NP-Complete.

Circle one:

True

False

Unknown

Justification (≤ 15 words):

(d) If a function A in NP has an exponential lower bound (e.g., requires $\Omega(2^n)$ time to compute), then no function in NP-Complete can be computed in polynomial time.

Circle one:

True

False

Unknown

Justification (≤ 3 sentences):

(e) If a function $Q : \{0, 1\}^* \rightarrow \{0, 1\}$ is computable, the function \overline{Q} is computable where $\forall x \in \{0, 1\}^* : \overline{Q}(x) = \text{NOT}(Q(x))$.

Circle one:

True

False

Unknown

Justification (≤ 3 sentences):

Proving Uncomputability

2. In this question, your goal is to show that the function $CELL_{15}$ defined below is uncomputable.

Input: A string w that describes a Turing Machine.

Output: **1** if the machine described by w would write a 1 on the fifteenth cell on its tape when executed on a tape that is initially all blank. Otherwise, **0**.

That is, a machine which computes $CELL_{15}$ outputs **1** when the input describes a Turing Machine which, when run on a blank tape, at some points writes the symbol 1 to the tape cell at index 15 (counting from the start-of-tape symbol at index 0).

(a) Which strategy would show that $CELL_{15}$ is uncomputable? (Circle one, no explication needed.)

Use a machine that computes
 $CELL_{15}$ to compute $HALTS$.

Use a machine that computes $HALTS$ to
compute $CELL_{15}$.

(b) Employ the strategy you chose in the previous question to show that $CELL_{15}$ is uncomputable.

Countable, Uncountable, Unknown

3. For each set described below, indicate whether its cardinality is *Countable*, *Uncountable*, or *Unknown* (not determined by the question if it is countable or uncountable). Circle one option and give a proof of your answer.

(a) The set of all grades that students will get on the final exam.

Countable

Uncountable

Unknown

Proof:

(b) The set of NAND circuits that compute *XOR*.

Countable

Uncountable

Unknown

Proof:

(c) The set of of all uncomputable languages.

Countable

Uncountable

Unknown

Proof:

Always, Sometimes, Never

4. For a function $f : \{0, 1\}^{3120} \rightarrow \{0, 1\}$ that can be implemented by a NAND circuit with s gates, which of the statements that follow would be *Always True*, *Possibly True* (meaning there are some functions f for which the statement is true and others for which it is false), or *Never True* (circle one option). Give a brief statement to justify your answer.

(a) f is computable

Always True

Possibly True

Never True

Justification (≤ 5 words):

(b) $f \in \text{NP}$

Always True

Possibly True

Never True

Justification (≤ 5 words):

(c) f can be implemented using 3120 NAND gates.

Always True

Possibly True

Never True

Justification (≤ 3 sentences):

Induction

5. Define the function $\text{ALT}_n : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ such that for a string $w \in \{0, 1\}^{2n}$ we say that $\text{ALT}_n(w) = 1$ provided $w \in (01)^*$. We could compute ALT_1 using the following straightline program:

```
def ALT1(x1, x2):  
    diff = XOR(x1, x2)  
    return AND(x2, diff)
```

We could then implement ALT_n as follows:

```
def ALTn(x1, x2, ..., x2n):  
    diff = XOR(x1, x2)  
    first = AND(x2, diff)  
    rest = ALTn(x3, ..., x2n)  
    return AND(first, rest)
```

Suppose we have similarly implemented ALT_{n-1} , ALT_{n-2} , etc., (and all other dependent subroutines).

Show that the number of NAND gates needed to represent a circuit for ALT_n is no more than $10n$ gates (hint: XOR requires 4 NAND gates and AND requires 3 NAND gates).

Complexity Classes

6. For an arbitrary given function A , for each of the complexity classes below, describe a way to prove that A belongs to the given class.

(a) P

(b) NP

(c) NP-Hard

(d) NP-Complete

(e) $O(n^2)$ (where n is the length of the input to A)

(f) $\Theta(1)$

(g) $\Omega(n)$ (where n is the length of the input to A)

7. Prove the following: If a function A in NP has an exponential lower bound (e.g. requires $\Omega(2^n)$ time to compute), then no language in NP-Complete can be computed in polynomial time.

Regular Expressions and Automata

8. For the following 4 sub-problems you will be asked to get both a regular expression and a finite state automaton for two different languages.

(a) Draw a finite state automaton (either an NFA or DFA) for the language:

$$\{x \in \{0, 1\}^* \mid x \text{ as interpreted as a binary representation of a natural number is odd}\}$$

(note that the empty string is a binary representation of 0, which is even).

(b) Give a regular expression for the language:

$$\{x \in \{0, 1\}^* \mid x \text{ as interpreted as a binary representation of a natural number is odd}\}$$

(note that the empty string is a binary representation of 0, which is even).

(c) Draw a finite state automaton (either an NFA or DFA) for the language: $\text{XOR} : \{0, 1\}^* \rightarrow \{0, 1\}$. In other words, the language $\{x \in \{0, 1\}^* \mid \text{XOR}(x) = 1\}$.

(d) Give a regular expression for the language: $\text{XOR} : \{0, 1\}^* \rightarrow \{0, 1\}$. In other words, the language $\{x \in \{0, 1\}^* \mid \text{XOR}(x) = 1\}$.

Models

- 9.** List the essential things that are required to define a model of computing.
- 10.** Describe how to show that two models of computing are equivalent.
- 11.** A Turing Machine's configuration contains all the information needed to describe the current status of its computation (i.e., if I paused my computation then wrote the configuration down, I could resume the computation using what I had written). List all the necessary components of a Turing Machine's configuration.

Asymptotics

12. Let $f(n) = 8n^{4.5}$ and $g(n) = 5n^5$, which of the following are true? Support your answer to each part with a convincing argument.

(a) $f \in O(g)$

(b) $f \in \Omega(g)$

(c) $f \in \Theta(g)$