CS 456/656 Fall 2025 Answers to Second Examination March 12

The entire test is 275 points.

1. True or False (5 points each). T = true, F = false, and O = open, meaning that the answer is not known science at this time.

   (i) **F** Every subset of a regular language is regular.

   (ii) **O** There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a binary numeral.

   (iii) **O** There exists a $\mathcal{P}$-TIME algorithm which finds a maximum independent set in any graph.

   (iv) **O** $\mathcal{NC} = \mathcal{P}$.

   (v) **O** $\mathcal{P} = \mathcal{NP}$.

   (vi) **T** The language consisting of all satisfiable Boolean expressions is known to be $\mathcal{NP}$-complete.

   (vii) **F** The context-free grammar equivalence problem is decidable.

   (viii) **T** For any two languages L_1 and L_2, if L_1 is $\mathcal{NP}$-complete and there is a recursive reduction of L_1 to L_2, then L_2 must be $\mathcal{NP}$–hard.

   (ix) **O** If L is $\mathcal{NP}$ and also co-$\mathcal{NP}$, then L must be $\mathcal{P}$.

   (x) **T** If L is $\mathcal{RE}$ and also co-$\mathcal{RE}$, then L must be decidable.

   (xi) **T** There exists a mathematical proposition which is true, but can be neither proved nor disproved.

   (xii) **F** For every real number x, there is a C++ program which runs forever, writing the string of decimal digits of x.

   (xiii) **F** 2-SAT is known to be $\mathcal{NP}$-complete.

   (xiv) **T** 3-SAT is known to be $\mathcal{NP}$-complete.

   (xv) **T** The complement of any $\mathcal{P}$-TIME language is $\mathcal{P}$-TIME.

   (xvi) **F** The complement of any undecidable language is undecidable.

   (xvii) **T** A language L is recursively enumerable if and only if there is a machine which accepts L.

   (xviii) **F** If $f(n) = \Theta(g(n))$ and g is a recursive (computable) function, then f must be recursive.

2. [5 points] Fill in the blank. A language L is **decidable** if and only if there is a machine which enumerates L in canonical order.

3. Which of these sets are countable? (5 points each) Write **T** for countable, **F** for uncountable, and **O** if the answer is not known.

   T The set of rational numbers.

   F The set of real numbers.

   T The set of recursive functions from $\mathcal{N}$ to $\mathcal{N}$.

   T The set of recursive real numbers.

   F The set of undecidable binary languages.

   T The set of recursively enumerable binary languages.

4. [10 points] State the Church Turing thesis.

   If a computation can be done by any machine, that computation can be done by some Turing machine.

5. [10 points] State the pumping lemma for regular languages.

   For any regular language L
   There exists a number p such that
   For any string $w \in L$ of length at least p
   There exist strings x, y, and z such that the following four statements hold:
   1. $w = xyz$.
   2. $|xy| \leq p$.
   3. $|y| \geq 1$.
   4. For any integer $i \geq 0$ $xy^iz \in L$.

6. [10 points] What is a recursive real number? Give a definition.

   There is more than one definitions. Any one of them is accepted as the correct answer. Here are three we covered in class. Let x be a real number.

   (i) x is recursive if and only if there is a machine which runs forever, writing the digits of the decimal (or binary, or any other base) expansion of x.

   (ii) x is recursive if and only if there is a computable function f such that $f(n)$ is the n^{th} decimal (or binary, or whatever) expansion of x.

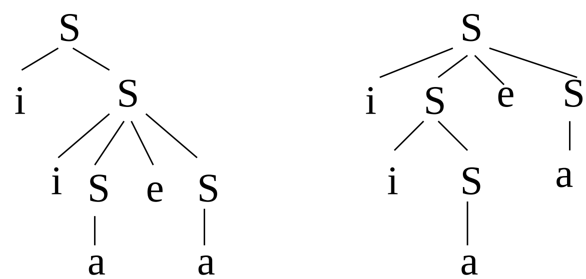
   (iii) x is recursive if and only if the set of fractions whose values are less than x is a decidable language.

7. [10 points] What does "$\mathcal{NP}$–hard" meain?

Informally, it means "at least as hard as an $\mathcal{NP}$–complete roblem." More formally, a language H is said to be $\mathcal{NP}$–hard if every $\mathcal{NP}$ language is $\mathcal{P}$-TIME reducible to H.

8. [10 points] Show that the context-free grammar given below is ambiguous by writing two different parse trees for the same string.

$S \to iS$
$S \to iSeS$
$S \to a$



9. [10 points] Give an unambiguous context-free grammar for the language of all palindromes over $\{a, b\}$

$S \to aSa$
$S \to bSb$
$S \to a$
$S \to b$
$S \to \lambda$

10. [10 points] Name a very important practical consequence if it became known that $\mathcal{P} = \mathcal{NP}$.

That would imply there exist no 1-way functions. (A 1-way function is a function f so that $f(n)$ is computable in polynomial time, but where, given $f(n)$, it is impossible to compute n in polynomial time. Cryptography based on 1-way funtions would then be breakable in polynomial time, In particular, RSA encryption, which is widely used, would be breakable in polynomial time.

11. [20 points] Prove that the halting problem is undecidable.

Suppose the halting problem is decidable. Let D be the following program:

Read a program P
If P halts with input P
    Loop forever
Else
    Halt

If D halts with input D, then D with input D loops forever, contradiction. If D does not halt with input D, then D with input D halts, contradiction. We conclude that there is no algorithm which solves the halting problem.

12. [20 points] Give a polynomial time reduction of the subset sum problem to Partition. An instance of the subset sum problem is a sequence of positive numbers $\sigma = x_1, x_2, \ldots, x_n$ together with a number K. A solution of that instance is a subsequence of σ whose terms total K. An instance of the partion problem is a sequence of numbers $\tau = y_1, y_2, \ldots, y_m$ and a solution to that instance is a partition of the terms of τ into two subsequences of equal total; alternatively stated, a subsequence of total half the total of the terms of τ.

Given instance σ, K of the subset sum problem, let S be the sum of the terms of σ. We map that instance to the sequence $\tau = x_1, x_2, \ldots x_n, K+1, S-K+1$. Then σ, K has a solution if and only if τ has a solution.

13. [20 points] Correctly state the verification definition of the class $\mathcal{NP}$.

A language L is $\mathcal{NP}$ if and only if there exists a number k and a program V such that
1. For every string $w \in L$ there exists a string c such that V accepts (w, c) in polynomial time.
2. For any string $w \notin L$ and any string c, V does not accept (w, c).

14. [20 points] Sketch a PDA for the language L of all strings over $\{a, b\}$ which have equal numbers of each symbol.

On the examination, I did not specify that the PDA had to be deterministic. The left figure shows a PDA which is not deterministic, while the right figure shows a DPDA. In both cases, the machine accepts L.