University of Nevada, Las Vegas Computer Science 456/656 Spring 2020

Practice Examination for April 30, 2020

Updated Fri Apr 24 13:55:39 PDT 2020

The entire practice examination is 480 points.

The current closure order extends to April 30. Thus the exam will be take-home.

1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time. In the questions below, $\mathcal{P}$ and $\mathcal{NP}$ denote $\mathcal{P}$-TIME and $\mathcal{NP}$-TIME, respectively.

   (i) **F** Let L be the language over $\{a, b, c\}$ consisting of all strings which have more a's than b's and more b's than c's. There is some PDA that accepts L.

   (ii) **T** The language $\{a^n b^n \mid n \geq 0\}$ is context-free.

   (iii) **F** The language $\{a^n b^n c^n \mid n \geq 0\}$ is context-free.

   (iv) Deleted.

   (v) **T** The intersection of any three regular languages is context-free.

   (vi) **T** If L is a context-free language over an alphabet with just one symbol, then L is regular.

   (vii) There is a deterministic parser for any context-free grammar. **F**

   (viii) **T** The set of strings that your high school algebra teacher would accept as legitimate expressions is a context-free language.

   (ix) **T** Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

   (x) **T** The problem of whether a given string is generated by a given context-free grammar is decidable.

   (xi) **T** If G is a context-free grammar, the question of whether $L(G) = \emptyset$ is decidable.

   (xii) **F** Every language generated by an unambiguous context-free grammar is accepted by some DPDA.

   (xiii) **T** The language $\{a^n b^n c^n d^n \mid n \geq 0\}$ is in the class $\mathcal{P}$-TIME.

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

   (xv) **F** Every undecidable problem is $\mathcal{NP}$-complete.

   (xvi) **F** Every problem that can be mathematically defined has an algorithmic solution.

   (xvii) **F** The intersection of two undecidable languages is always undecidable.

   (xviii) **T** Every $\mathcal{NP}$ language is decidable.

   (xix) **T** The intersection of two $\mathcal{NP}$ languages must be $\mathcal{NP}$.

(xx) Deleted.

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

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

(xxiii) **O** $\mathcal{NP} = \mathcal{P}$-SPACE

(xxiv) **O** $\mathcal{P}$-SPACE = EXP-TIME

(xxv) **O** EXP-TIME = EXP-SPACE

(xxvi) **F** NC = EXP-SPACE

(xxvii) **T** The traveling salesman problem (TSP) is $\mathcal{NP}$-complete.

(xxviii) **T** The knapsack problem is $\mathcal{NP}$-complete.

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

(xxx) **T** The Boolean Circuit Problem is in $\mathcal{P}$.

(xxxi) **T** The Boolean Circuit Problem is in $\mathcal{NC}$.

(xxxii) Deleted.

(xxxiii) **T** The language consisting of all strings over $\{a, b\}$ which have more a's than b's is context-free.

(xxxiv) **T** 2-SAT is $\mathcal{P}$-TIME.

(xxxv) **O** 3-SAT is $\mathcal{P}$-TIME.

(xxxvi) **T** Primality is $\mathcal{P}$-TIME.

(xxxvii) **T** There is a $\mathcal{P}$-TIME reduction of the halting problem to 3-SAT.

(xxxviii) **T** There is a $\mathcal{P}$-TIME reduction of the partition problem to 3-SAT.

(xxxix) **T** Every context-free language is in $\mathcal{P}$.

(xl) **T** Every context-free language is in $\mathcal{NC}$.

(xli) **T** Addition of binary numerals is in $\mathcal{NC}$.

(xlii) **O** Every context-sensitive language is in $\mathcal{P}$.

(xliii) **F** Every language generated by a general grammar is recursive.

(xliv) **F** The problem of whether two given context-free grammars generate the same language is decidable.

(xlv) **T** The language of all fractions (using base 10 numeration) whose values are less than π is decidable. (A *fraction* is a string. "314/100" is in the language, but "22/7" is not.)

(xlvi) **T** There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a unary ("caveman") numeral.

(xlvii) **F** For any two languages L_1 and L_2, if L_2 is undecidable and there is a recursive reduction of L_1 to L_2, then L_1 must be undecidable.

(xlviii) **F** If P is a mathematical proposition that can be written using string of length n, and P has a proof, then P must have a proof whose length is $O(2^{2^n})$.

As you may have learned, there is a formal language which can be used to write any mathematical proposition as well as any proof of any mathematical proposition, and an algorithm exists that can check the correctness of such a proof. In 1978, Jack Milnor https://en.wikipedia.org/wiki/John_Milnor told me that in the future no proof will be accepted unless it can be verified by a computer.

(xlix) Deleted.

(l) **F** Every bounded function is recursive.

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

(lii) **T** Recall that if $\mathcal{L}$ is a class of languages, co-$\mathcal{L}$ is defined to be the class of all languages that are not in $\mathcal{L}$. Let $\mathcal{RE}$ be the class of all recursively enumerable languages. If L is in $\mathcal{RE}$ and also L is in co-$\mathcal{RE}$, then L must be decidable.

(liii) **T** Every language is enumerable.

(liv) **F** If a language L is undecidable, then there can be no machine that enumerates L.

(lv) **T** There is a non-recursive function which grows faster than any recursive function.

(lvi) **T** There exists a machine[1] that runs forever and outputs the string of decimal digits of π (the well-known ratio of the circumference of a circle to its diameter).

There are several known series that converge to π. Using one of these, it is easy to write the program. I've done it. The only problem is that any physical machine will eventually run out of memory. Here is one series: $\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots$ converges to $\frac{\pi}{4}$.

(lvii) **T** There is a machine which outputs the correct answer to the $\mathcal{P} = \mathcal{NP}$ question.

The correct answer to that problem consists of just one string, which is either "0" or "1" Design a machine M_1 which simply outputs "1" when it is turned on, and a machine M_0 which simply outputs "0" when you turn it one. One of those two machines outputs the correct answer to the $\mathcal{P} = \mathcal{NP}$ question, the machine exists. The fact that we don't know whether it's M_0 or M_1 is irrelevant.

(lviii) **F** For every real number x, there exists a machine that runs forever and outputs the string of decimal digits of x.

There are uncountably real numbers, and there are only countably many machines.

[1]As always in automata theory, "machine" means abstract machine, a mathematical object whose memory and running time are **not** constrained by the size and lifetime of the known (or unknown) universe, or any other physical laws. If we want to discuss the kind of machine that exists (or could exist) physically, we call it a "physical machine."

(lix) Deleted

(lx) **O Rush Hour**, the puzzle sold in game stores everywhere, generalized to a board of arbitrary size, is $\mathcal{NP}$

Rush Hour has been known to be $\mathcal{P}$-SPACE complete since 2002.

(lxi) **O** The regular expression equivalence problem is $\mathcal{NP}$.

The problem is known to be $\mathcal{P}$-SPACE complete.

(lxii) **F** The intersection of two context-free languages is always context-free.

(lxiii) **T** The intersection of two regular languages is always regular.

(lxiv) **T** The intersection of a context-free language and a regular language is always context-free.

If M_1 is a PDA which accepts L_1 and M_2 is a DFA which accepts L_2, the Cartesian product machine $M_1 \times M_2$ has only one stack, and can thus be rewritten as a PDA, and accepts $L_1 \cap L_2$.

(lxv) **F** $\left\{a^i b^j c^i d^j \ : \ i, j \geq 0\right\}$ is context-free.

(lxvi) **T** $\left\{a^i b^i c^j d^j \ : \ i, j \geq 0\right\}$ is context-free.

(lxvii) **T** $\left\{a^i b^j c^j d^i \ : \ i, j \geq 0\right\}$ is context-free.

2. [10 points] Fill in the blank. If L_1 is undecidable and if R is a reduction of L_1 to L_2 and if R is **recursive**, then L_2 is undecidable.

3. [20 points] State the pumping lemma for context-free languages.

   For any context-free language L, there exists a positive integer p, called a pumping length of L, such that for any $w \in L$ such that $|w| \geq p$ there exist strings u, v, x, y, z such that the following four conditions hold:

   (a) $w = uvxyz$
   (b) $|vxy| \leq p$
   (c) $|vy| \geq 1$
   (d) for any integer $i \geq 0$, $uv^ixy^iz \in L$

4. [20 points] Use the pumping lemma for context-free languages to prove that the language $L = \{a^n b^n c^n\}$ is not context-free.

   Proof: By contradiction. Assume L is context-free. Then by the pumping lemma L has a pumping length $p \geq 1$.

   Let $w = a^p b^p c^p$ Then $w \in L$ and $|w| = 3p \geq p$. Pick string u, v, x, y, z such that the four final conditions given in the pumping lemma hold.

   Any substring of w which contains at least one a and one c must have length at least $p + 2$. Thus, vxy either contains no a or contains no c. Without loss of generality, vxy contains no a.

Pick $i = 0$. Then $uv^ixy^iz = uxz \in L$. Since neither v nor y contains any a, the number of a's in uxz must be the same as the number of a's in w, which is p. Since $uvz \in L$, $\{uvz\} = 3p$. It follos that v and y are both the empty string, contradiction.∎

5. [20 points] Prove that a language is recursively enumerable if and only if it is accepted by some machine.

   If a language L is recursively enumerable, it is accepted by some machine.

   Proof: There is a machine which enumerates L. Let $w_1, w_2, \ldots$ be that enumeration. Let P be the following program:

   read(w)
   for all i starting from 1
   if($w = w_i$) output "1"

   If P outputs 1, then $w = w_i \in L$ for some i, hence $w \in L$. Conversely, if $w \in L$, then $w = w_i$ for some i because $w_1, \ldots$ is an enumeration of L, hence P outputs 1. Thus P accepts L. ∎

   If L is accepted by some machine, then L is recursively enumerable.

   Proof: Let M be a machine which accepts L. Let Σ be the alphabet of L. Let $w_1, w_2, \ldots$ be an enumeration of Σ^*. Let P be the following program:

   for $t = 1$ to ∞
       for $i = 1$ to t
           if (M accepts w_i within t steps)
               output w_i

   P runs forever, but the inner loop is finite for each t. Only strings accepted by M are output. Conversely, if w is accepted by M, then $w = w_i$ for some i, and M accepts w in k steps for some k. Then w will be output during the t^{th} iteration of the outer loop, where $t = \max\{i, k\}$. ∎

6. [20 points] Prove that the halting problem is undecidable.

7. [20 points] Give a definition of each of these $\mathcal{NP}$-complete languages/problems.

   (a) SAT = the set of all satisfiable Boolean expressions.

   (b) 3-SAT = the set of all satisfiable Boolean expressions which are in conjunction normal form where each clause has three terms. An expression must be the conjunction of clauses, and each clause must be the disjunction of three terms, and each term must be either a variable or the negation of a variable.

   (c) The independent set problem is whether a given graph G has an independent set of k vertices for some given k. We say that a set of vertices I is independent if no two members of I have an edge in common.

   (d) The subset sum problem is whether there is a subset of a given set of weighted items whose total weight equals a given number..

   (e) The partition problem is whether a given set of weighted items can be divided into two subsets of equal total weight.

8. [20 points] Give a general (unrestricted) grammar for the language consisting of all strings of 1's of length a power of 2, that is, $\left\{1^{2^n} : n \geq 0\right\}$

   We will use variables S,L,R,D where S is the start symbol and the productions are
   S$\rightarrow$L1R
   L$\rightarrow$LD
   D1$\rightarrow$11D
   DR$\rightarrow$R
   L$\rightarrow \lambda$
   R$\rightarrow \lambda$

9. [20 points] Give one of these polynomial time reductions (your choice).

   (a) 3-SAT to Independent Set.

   (b) Independent Set to Subset Sum

   (c) Subset Sum to Partition

10. [20 points] Prove that the context-free grammar equivalence problem is co-RE.

I have already made the proofs of 6, 9, and 10 available. Review them.