Computer Science 456/656 Trove of True/False Questions from Earlier Semesters

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) ______ 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) ______ The language $\{a^n b^n \mid n \geq 0\}$ is context-free.

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

   (iv) ______ The language $\left\{a^i b^j c^k \mid j = i + k\right\}$ is context-free.

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

   (vi) ______ 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.

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

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

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

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

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

   (xiii) ______ The language $\{a^n b^n c^n d^n \mid n \geq 0\}$ is recursive.

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

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

   (xvi) ______ Every undecidable problem is $\mathcal{NP}$-complete.

   (xvii) ______ Every problem that can be mathematically defined has an algorithmic solution.

   (xviii) ______ The intersection of two undecidable languages is always undecidable.

   (xix) ______ Every $\mathcal{NP}$ language is decidable.

   (xx) ______ The intersection of two $\mathcal{NP}$ languages must be $\mathcal{NP}$.

   (xxi) ______ The intersection of two $\mathcal{NP}$-complete languages must be $\mathcal{NP}$-complete.

(xxii) ______ $\mathcal{NC} = \mathcal{P}$.

(xxiii) ______ $\mathcal{P} = \mathcal{NP}$.

(xxiv) ______ $\mathcal{NP} = \mathcal{P}$-SPACE

(xxv) ______ $\mathcal{P}$-SPACE = EXP-TIME

(xxvi) ______ EXP-TIME = EXP-SPACE

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

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

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

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

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

(xxxii) ______ If L_1 and L_2 are undecidable langugages, there must be a recursive reduction of L_1 to L_2.

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

(xxxiv) ______ The language consisting of all strings over $\{a, b\}$ which have more a's than b's is context-free.

(xxxv) ______ 2-SAT is $\mathcal{P}$-TIME.

(xxxvi) ______ 3-SAT is $\mathcal{P}$-TIME.

(xxxvii) ______ Primality is $\mathcal{P}$-TIME.

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

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

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

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

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

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

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

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

(xlviii) ______ 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.

(xlix) 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.

______ 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 $O(2^{2^n})$.

(l) ______ If L is any $\mathcal{NP}$ language, there must be a $\mathcal{P}$–TIME reduction of the partition problem to L.

(li) ______ Every bounded function is recursive.

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

(liii) ______ 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.

(liv) ______ Every language is enumerable.

(lv) ______ If a language L is undecidable, then there can be no machine that enumerates L.

(lvi) ______ There exists a mathematical proposition that can be neither proved nor disproved.

(lvii) ______ There is a non-recursive function which grows faster than any recursive function.

(lviii) ______ 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).

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

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

(lxi) ______ There is a polynomial time algorithm which determines whether any two regular expressions are equivalent.

(lxii) ______ If two regular expressions are equivalent, there is a polynomial time proof that they are equivalent.

[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."