Computer Science 456/656 Spring 2018

1. Turn this problem in on Wednesday, February 28, 2018.

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

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

   (iii) _____ 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.

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

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

   (vi) _____ Every undecidable problem is $\mathcal{NP}$-complete.

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

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

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

   (x) _____ The intersection of any three regular languages is context-free.

   (xi) _____ Every problem that can be mathematically defined has an algorithmic solution.

   (xii) _____ The intersection of two undecidable languages is always undecidable.

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

   (xiv) _____ Every $\mathcal{NP}$ language is decidable.

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

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

   (xvii) _____ $\mathcal{NC} = \mathcal{P}$.

   (xviii) _____ $\mathcal{P} = \mathcal{NP}$.

   (xix) _____ $\mathcal{NP} = \mathcal{P}$-SPACE

   (xx) _____ $\mathcal{P}$-SPACE = EXP-TIME

   (xxi) _____ EXP-TIME = EXP-SPACE

   (xxii) _____ There is a deterministic parser for any context-free grammar.

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

(xxiv) _____ The knapsack problem is $\mathcal{NP}$-complete.

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

(xxvi) _____ The Boolean Circuit Problem is in $\mathcal{P}$.

(xxvii) _____ The Boolean Circuit Problem is in $\mathcal{NC}$.

Turn this problems in on Monday March 5.

2. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

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

   (ii) _____ The language $\{\epsilon\}$ consisting of just one string, the empty string, is $\mathcal{NP}$-complete. **Warning.** Think long and hard about this problem before you answer. It's not as easy as it seems.

   (iii) _____ The empty language is $\mathcal{NP}$-complete. **Warning.** Think long and hard about this problem before you answer. It's not as easy as it seems.

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

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

   (vi) _____ 2-SAT is $\mathcal{P}$-TIME.

   (vii) _____ 3-SAT is $\mathcal{P}$-TIME.

   (viii) _____ Primality is $\mathcal{P}$-TIME.

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

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

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

   (xii) _____ Every language generated by a general grammar is recursive.

   (xiii) _____ Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

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

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

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

(xvii) _____ If G is a context-free grammar, the question of whether $L(G) = \Sigma^*$ is decidable, where Σ is the terminal alphabet of G.

(xviii) _____ The set of all fractions whose values are less than π is decidable. (A *fraction* is a string, defined to be a non-empty string of decimal digits followed by a slash followed by a non-empty string of decimal digits, such as "3/42")

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

(xx) _____ 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.

(xxi) _____ 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.

(xxii) 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. In 1978, Jack 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 $O(2^{2^n})$.

(xxiii) _____ There is a $\mathcal{P}$-TIME reduction of the context-free grammar membership problem to the halting problem.

(xxiv) _____ Every bounded function is recursive.

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

(xxvi) _____ 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.

(xxvii) _____ Every language is enumerable.

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

(xxix) _____ There exists a mathematical proposition that can be neither proved nor disproved.

(xxx) _____ There is a non-recursive function which grows faster than any recursive function.

(xxxi) _____ 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).

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

[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."