Computer Science 456/656. Problem 1 of Assignment 9, Fall 2013.

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

   (i) ______ 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).

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

   (iii) ______ If a language has an unambiguous context-free grammar, then there must be some DPDA that accepts it.

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

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

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

   (vii) ______ 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.

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

   (ix) ______ The problem of whether a given context-sensitive grammar generates a given string is in the class $\mathcal{NP}$.

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

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

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

   (xiii) ______ The problem of whether a given context-free grammar generates all strings is decidable.

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

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

[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."

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

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

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

(xix) ______ **(Warning: this one is hard.)** If f is any function on positive integers, there must be a recursive function g such that $f(n) = O(g(n))$.

(xx) 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.

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

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

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

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

(xxv) ______ Integer programming is $\mathcal{NP}$-hard.

(xxvi) ______ Linear programming is $\mathcal{NP}$-hard.

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

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

(xxix) ______ The traveling salesman problem is $\mathcal{NP}$-hard.

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

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

(xxxii) ______ There exists a $\mathcal{P}$-TIME algorithm which finds a maximum independent set in any graph G.

(xxxiii) ______ There exists a $\mathcal{P}$-TIME algorithm which finds a maximum independent set in any acyclic graph G.

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

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

(xxxvi) ______ The set of all fractions whose values are less than π is decidable. (Think of a fraction as a string of digits, followed by a slash, followed by another string of digits.)

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

(xxxviii) ______ Every bounded function is recursive.

(xxxix) ______ For any two languages L_1 and L_2, if L_1 is $\mathcal{NP}$-complete, L_2 is $\mathcal{NP}$, and there is a polynomial time reduction of L_1 to L_2, then L_2 must be $\mathcal{NP}$-complete.

(xl) ______ For any two languages L_1 and L_2, if L_2 is $\mathcal{NP}$-complete, L_1 is $\mathcal{NP}$, and there is a polynomial time reduction of L_1 to L_2, then L_1 must be $\mathcal{NP}$-complete.

(xli) ______ If P is a mathematical proposition that can be stated using n binary bits, and P has a proof, then P must have a proof whose length is $O(2^{2^n})$.

(xlii) ______ Let L be the language consisting of just one string of length 1, defined as follows:

$$L = \begin{cases} \{1\} \text{ if } \mathcal{P} = \mathcal{NP} \\ \{0\} \text{ if } \mathcal{P} \neq \mathcal{NP} \end{cases}$$

Then L is undecidable.

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

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

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

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