Answers to True/False Questions

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

   (v) **T** The intersection of any two regular languages is regular.

   (vi) **T** The intersection of any regular language with any context-free language is context-free.

   (vii) **F** The intersection of any two context-free languages is context-free.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(xxiii) **F** If L_1 and L_2 are known to be $\mathcal{NP}$-complete languages and $L_1 \cap L_2$ is not empty, then $L_1 \cap L_2$ is known to be $\mathcal{NP}$-complete.

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

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

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

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

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

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

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

(xxxi) **F** EXP-TIME = $\mathcal{P}$-TIME.

(xxxii) **F** EXP-SPACE = $\mathcal{P}$-SPACE.

(xxxiii) **T** The traveling salesman problem (TSP) is known to be $\mathcal{NP}$-complete.

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

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

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

(xxxvii) **F** If L_1 and L_2 are undecidable langugages, there must be a recursive reduction of L_1 to L_2.

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

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

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

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

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

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

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

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

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

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

(xlviii) **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.)

(xlix) **T** Any context-free language over the unary alphabet is regular.

(l) **F** Any context-sensitive language over the unary alphabet is regular.

(li) **F** Any recursive language over the unary alphabet is regular.

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

(liii) **T** 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.

(liv) **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.

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

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

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

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

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

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

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

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

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

(lxiv) **T** There exists a machine 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).

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

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

(lxvii) **O** There is a polynomial time algorithm which determines whether any two regular expressions are equivalent.

(lxviii) **O** If two regular expressions are equivalent, there is a polynomial time proof that they are equivalent.

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

(lxx) **T** Every subset of any enumerable set is enumerable.

(lxxi) **T** The computer language Pascal has Turing power.

(lxxii) **T** Computing the square of an integer written in binary notation is an $\mathcal{NC}$ function.

(lxxiii) **T** If L is any $\mathcal{P}$-TIME language, there is an $\mathcal{NC}$ reduction of L to the Boolean circuit problem.

(lxxiv) **T** If an abstract Pascal machine can perform a computation in polynomial time, there must be some Turing machine that can perform the same computation in polynomial time.

(lxxv) **T** The binary integer factorization problem is co-$\mathcal{NP}$.

(lxxvi) **T** There is a polynomial time reduction of the subset sum problem to the binary numeral factorization problem.

(lxxvii) **O** There is a polynomial time reduction of the binary numeral factorization problem to the subset sum problem.

(lxxviii) **F** For any real number x, the set of fractions whose values are less than x is $\mathcal{RE}$.

(lxxix) **T** For any recursive real number x, the set of fractions whose values are less than x is recursive.

(lxxx) **F** The union of any two deterministic context-free languages must be a DCFL.

(lxxxi) **F** The intersection of any two deterministic context-free languages must be a DCFL.

(lxxxii) **F** The complement of any DCFL must be a DCFL.

(lxxxiii) **T** The membership problem for a DCFL is in the class $\mathcal{P}$-TIME.

(lxxxiv) **T** Every finite language is decidable.

(lxxxv) **T** Every context-free language is in Nick's class.

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

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

(lxxxviii) **T** The complement of any $\mathcal{P}$–SPACE language is $\mathcal{P}$–SPACE.

The *jigsaw puzzle problem* is, given a set of various polygons, and given a rectangular table, is it possibe to assemble those polygons to exactly cover the table?

The *furniture mover's problem* is, given a room with a door, and given a set of objects outside the room, it is possible to move all the objects into the room through the door?

(lxxxix) **T** The jigsaw puzzle problem is known to be $\mathcal{NP}$ complete.

(xc) **F** The jigsaw puzzle problem is known to be $\mathcal{P}$–SPACE complete.

(xci) The furniture mover's problem is known to be $\mathcal{NP}$ complete.

(xcii) **T** The furniture mover's problem is known to be $\mathcal{P}$–SPACE complete.

(xciii) **T** The complement of any recursive language is recursive.

(xciv) **T** The complement of any undecidable language is undecidable.

(xcv) **F** Every undecidable language is either $\mathcal{RE}$ or co-$\mathcal{RE}$.

(xcvi) **T** For any infinite countable sets A and B, there is a 1-1 correspondence between A and B.

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

(xcviii) **T** Every $\mathcal{NP}$ language is reducible to the independent set problem in polynomial time.

(xcix) **T** If a Boolean expression is satisfiable, there is a polynomial time proof that it is satisfiable.

(c) **T** The general sliding block problem is $\mathcal{P}$–SPACE complete.

(ci) **T** The regular epression equivalence problem is $\mathcal{P}$–SPACE complete.

(cii) **T** The context-sensitive membership problem is $\mathcal{P}$–SPACE complete.

(ciii) **T** The Post correspondence problem is undecidable.