True/False Questions

$\mathcal{P}$ means $\mathcal{P}$–TIME

$calN\mathcal{P}$ means $\mathcal{NP}$–TIME

$calR\mathcal{E}$ means recursively enumerable
$\mathcal{NC}$ means Nick's class.
If $\mathcal{C}$ is any class of languages, co-$\mathcal{C}$ means the class of all languages which are complements of languages in $\mathcal{C}$.
A *binary language* is a language over the binary alphabet $\{0,1\}$.
A *recursive* function is any function which can be computed by a machine.
A *recursive* real number is any real number whose n^{th} decimal digit is a recursive function of n.
A *fraction* is a string consisting of a numeral, followed by a slash, followed by another numeral.
The *unary* alphabet is $\{1\}$. Some books write $\{0\}$. But in any case, the unary alphabet has just one symbol.
Countable is just another word for *enumerable.*

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** Every subset of a regular language is regular.

   (ii) **T** The union of any two regular languages must be regular

   (iii) **T** The concatenation of any two regular languages must be regular

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

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

   (vi) **T** The complement of any regular language is regular.

   (vii) The Kleene closure of any regular language is regular. **T**

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

   (ix) **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.

   (x) **F** The union of any two context-free languages must be context-free

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

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

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

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

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

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

(xvii) **T** There is a deterministic parser for any context-free grammar.

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

(xix) **T** The Kleene closure of any context-free language is context-free.

(xx) **T** Every regular language is context-free.

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

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

(xxiii) **T** The problem of whether a given string is generated by a given context-free grammar is decidable. (Use the CYK algorithm.)

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

(xxv) **T** The Kleene closure of any $\mathcal{NP}$ langauge is $\mathcal{NP}$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(xlvi) **T** $\mathcal{NP}$-SPACE = $\mathcal{P}$-SPACE.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(lxi) **T** The language of all fractions (using base 10 numeration) whose values are less than π is decidable.

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

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

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

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

(lxvi) **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.

(lxvii) **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.

(lxviii) **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 $O(2^{2^n})$.

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

(lxx) **T** A language is decidable if and only if it is recursive.

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

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

(lxxiii) **T** A language is $\mathcal{RE}$ if and only if it is generated by a grammar.

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

(lxxv) **F** Every language is enumerable.

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

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

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

(lxxix) **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).

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

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

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

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

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

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

(lxxxvi) **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.

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

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

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

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

(xci) **F** For any recursive real number x, the set of fractions whose values are less than x is recursive (i.e., decidable).

(xcii) **T** The concatenation of any two deterministic context-free languages must be a DCFL.

(xciii) **T** The concatenation of any two context-free languages must be context-free

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

(xcv) **T** The membership problem for any CFL is in the class $\mathcal{P}$-TIME.

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

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

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

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

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

(ci) **T** The complement of any decidable language is decidable.

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

(ciii) **F** The complement of any $\mathcal{RE}$ language is $\mathcal{RE}$.

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?

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

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

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

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

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

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

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

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

(cxii) **F** The set of all binary languages is countable.

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

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

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

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

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

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

(cxix) **F** The Post correspondence problem is decidable.

(cxx) **F** The halting problem is decidable.