

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) T There exists a machine¹ 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) F For every real number x , there exists a machine that runs forever and outputs the string of decimal digits of x .
 - (iii) F If a language has an unambiguous context-free grammar, then there must be some DPDA that accepts it.
 - (iv) F The problem of whether two given context-free grammars generate the same language is decidable.
 - (v) T The problem of whether a given string is generated by a given context-free grammar is decidable.
 - (vi) T The language $\{a^n b^n c^n d^n \mid n \geq 0\}$ is recursive.
 - (vii) 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 .
 - (viii) T There exists a mathematical proposition that can be neither proved nor disproved.
 - (ix) O The problem of whether a given context-sensitive grammar generates a given string is in the class \mathcal{NP} .
 - (x) T The language $\{a^n b^n c^n \mid n \geq 0\}$ is in the class \mathcal{P} .
 - (xi) O There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a binary numeral.
 - (xii) F Every undecidable problem is \mathcal{NP} -complete.
 - (xiii) F The problem of whether a given context-free grammar generates all strings is decidable.
 - (xiv) T The language $\{a^n b^n \mid n \geq 0\}$ is context-free.
 - (xv) F The language $\{a^n b^n c^n \mid n \geq 0\}$ is context-free.

¹As 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) T The language $\{a^i b^j c^k \mid j = i + k\}$ is context-free.
- (xvii) T The intersection of any three regular languages is context-free.
- (xviii) F If a language L is undecidable, then there can be no machine that enumerates L .
- (xix) F (**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, $\text{co-}\mathcal{L}$ is defined to be the class of all languages that are not in \mathcal{L} .
 T Let \mathcal{RE} be the class of all recursively enumerable languages. If L is in \mathcal{RE} and also L is in $\text{co-}\mathcal{RE}$, then L must be decidable.
- (xxi) F Every problem that can be mathematically defined has an algorithmic solution.
- (xxii) F The intersection of two undecidable languages is always undecidable.
- (xxiii) O If L is \mathcal{NP} and also $\text{co-}\mathcal{NP}$, then L must be \mathcal{P} .
- (xxiv) T If L is a context-free language over an alphabet with just one symbol, then L is regular.
- (xxv) T Integer programming is \mathcal{NP} -hard.
- (xxvi) O Linear programming is \mathcal{NP} -hard.
- (xxvii) T Every \mathcal{NP} language is decidable.
- (xxviii) T The clique problem is \mathcal{NP} -complete.
- (xxix) T The traveling salesman problem is \mathcal{NP} -hard.
- (xxx) T The intersection of two \mathcal{NP} languages must be \mathcal{NP} .
- (xxxi) O The intersection of two \mathcal{NP} -complete languages must be \mathcal{NP} -complete.
- (xxxii) O There exists a \mathcal{P} -TIME algorithm which finds a maximum independent set in any graph G .
- (xxxiii) T There exists a \mathcal{P} -TIME algorithm which finds a maximum independent set in any acyclic graph G .
- (xxxiv) T If G is a context-free grammar, the question of whether $L(G) = \emptyset$ is decidable.

- (xxxv) F 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) T 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) T There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a unary numeral.
- (xxxviii) F Every bounded function is recursive.
- (xxxix) T 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) O 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) F 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) F 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.

- (xliv) T 2-SAT is \mathcal{P} -TIME.
- (xlv) O 3-SAT is \mathcal{P} -TIME.
- (xlv) F There is a \mathcal{P} -TIME reduction of the halting problem to 3-SAT.
- (xlvi) T There is a \mathcal{P} -TIME reduction of the context-free grammar membership problem to the halting problem.