`Thu Nov 10 11:33:49 PST 2011`

T/F/O: Known to be true, Known to be false, or Open.

1. **F** Every subset of a regular language is regular.
2. **F** Let L be the language over $\Sigma = \{a, b\}$ consisting of all strings of the form $a^m b^n$, where $m \geq n$. Then L is a regular language.
3. **T** The complement of every regular language is regular.
4. **T** If a language has an context-free grammar, then it is is accepted by some push-down automaton.
5. **F** If a language has an ambiguous context-free grammar, then it is is not accepted by any deterministic push-down automaton.
6. **F** There is a PDA that accepts all valid C++ programs.
7. **T** The intersection of any two regular languages is regular.
8. **T** The language consisting of all base 3 numerals for positive integers n such that $n \,\%\, 7 = 3$ is regular.
9. **F** The language consisting of all base 5 numerals for prime positive integers is regular.
10. **F** The intersection of any two context-free languages is context-free.
11. **T** The Kleene closure of every context-free language is context-free.
12. **F** If a language has an unambiguous context-free grammar, then it must be accepted by some deterministic push-down automaton.
13. **O** $\mathcal{P}$-TIME $= \mathcal{NP}$-TIME.
14. **F** If a language is not in the class $\mathcal{P}$-TIME, then it must be undecidable.
15. **F** The set of all mathematical statements which are provably true is decidable.
16. **T** The set of all mathematical statements which are provably true is recursively enumerable.
17. **T** If a language L is accepted by some NTM, then there is a TM which accepts L.
18. **T** The problem of whether a given grammar generates the empty language is decidable.
19. **F** The problem of whether a given grammar generates all strings over an alphabet Σ is decidable.
20. **T** Every language that can be parsed by an LALR parser can be accepted by some DPDA.
21. **T** If a real number x is the solution to a polynomial equation with integral coefficients, then there must be a TM that runs forever, writing the decimal digits of x.

22. **T** There exists a polynomial time algorithm which finds the factors of any positive integer, if the input integer is written in unary ("caveman") notation.

23. **F** The Boolean circuit problem is undecidable.

24. **T** The complement of every recursive language is recursive.

25. **F** The complement of every recursively enumerable language is recursively enumerable.

26. **T** Every language which is generated by an unrestricted grammar is recursively enumerable.

27. **T** The question of whether two context-free grammars generate the same language is undecidable.

28. **T** There exists some proposition which is true but which has no proof.

29. **T** The set of all binary numerals for prime numbers is in the class $\mathcal{P}$.

30. **F** If L_1 reduces to L_2 in polynomial time, and if L_1 is $\mathcal{NP}$, and if L_2 is $\mathcal{NP}$-complete, then L_1 must be $\mathcal{NP}$-complete.

31. **T** If L_1 reduces to L_2 in polynomial time, and if L_2 is $\mathcal{NP}$, and if L_1 is $\mathcal{NP}$-complete, then L_2 must be $\mathcal{NP}$-complete.

32. **F** Given any context-free grammar G and any string $w \in L(G)$, there is always a unique leftmost derivation of w using G.

33. **F** For any deterministic finite automaton, there is always a unique minimal non-deterministic finite automaton equivalent to it.

34. **F** Using multi-processors and other advanced technology, it is possible to design a machine which decides the halting problem.

35. **T** The question of whether two regular expressions are equivalent is decidable.

36. **T** Let $L = \{\langle M \rangle \mid M \text{ halts with no input}\}$. Then L is recursively enumerable.

37. **F** The complement of every context-free language is context-free.

38. **O** The complement of every $\mathcal{NP}$ language is $\mathcal{NP}$.

39. **T** The class of context-free languages is a subclass of $\mathcal{P}$.

40. **F** No language which has an ambiguous context-free grammar can be accepted by a DPDA.

41. **F** The question of whether a given Turing Machine halts with empty input is decidable.

42. **T** The class of languages accepted by non-deterministic finite automata is the same as the class of languages accepted by deterministic finite automata.

43. **F** If L_1 reduces to L_2 in polynomial time, L_1 is $\mathcal{NP}$, then L_2 must be $\mathcal{NP}$.

44. **T** If L_1 reduces to L_2 in polynomial time, L_2 is $\mathcal{NP}$, then L_1 must be $\mathcal{NP}$.

45. **T** Let $F(0) = 1$, and let $F(n) = 2^{F(n-1)}$ for $n > 0$. Then F is Turing-computable.

46. **T** Every language which is accepted by some non-deterministic machine is accepted by some deterministic machine.

47. **T** The language of all regular expressions over the binary alphabet is a regular language.

48. **T** Let π be the ratio of the circumference of a circle to its diameter. (That's the usual meaning of π you learned in second grade.) The problem of whether the n^{th} digit of π, for a given n, is equal to a given digit is decidable.

49. **T** There cannot exist any computer program that can decide whether any two C++ programs are equivalent.

50. **F** An undecidable language is necessarily $\mathcal{NP}$-complete.

51. **T** Every context-free language is in the class $\mathcal{P}$-TIME.

52. **F** Every function that can be mathematically defined is Turing computable.

53. **F** The language of all binary strings which are the binary numerals for prime numbers is context-free.

54. **F** Every bounded function from integers to integers is Turing-computable. (We say that f is *bounded* if there is some B such that $|f(n)| \leq B$ for all n.)

55. **F** The language of all palindromes over $\{0, 1\}$ is inherently ambiguous.

56. **F** Every context-free grammar can be parsed by some deterministic top-down parser.

57. **T** Every context-free grammar can be parsed by some non-deterministic top-down parser.

58. **F** Every context-free grammar can be parsed by some deterministic bottom-up parser.

59. **T** Every context-free grammar can be parsed by some non-deterministic bottom-up parser.

60. **F** Commercially available parsers cannot use the LALR technique, since most modern programming languages are not context-free.

61. **F** The Boolean satisfiability problem is undecidable.

62. **O** There is a parallel processor machine which can solve the boolean circuit problem in polylogarithmic time.

63. **T** The set of all binary numerals for prime numbers is in the class $\mathcal{P}$-TIME.

64. **T** If $\mathcal{P} = \mathcal{NP}$, then RSA coding is insecure.

65. **T** The Boolean satisfiability problem is $\mathcal{NP}$-complete.

66. Fill in the blanks.

(a) Name two classes of machines that accept the class of regular languages. **DFAs** and **NFAs**.

(b) Name one class of machines that accepts the class of context-free languages. **PDAs**.

(c) If a machine M is **deterministic**, there is at most one legal move M can make from any give configuration.

(d) An LALR parser outputs a derivation of its input string which corresponds to the **postorder** visitation of the internal nodes of its parse tree.

(e) If every string generated by a grammar G has a unique leftmost derivation using G, then we say that G is **unambiguous**.

(f) We are able to prove that every language accepted by an NTM is accepted by a TM. In this proof, we construct a TM that emulates the NTM using a bitstring which we call a **guide** string.

67. [30 points] Consider the NFA whose transition diagram is drawn below, where the input alphabet is $\{a, b, c\}$. Draw the transition diagram of an equivalent minimal DFA. Show your steps.

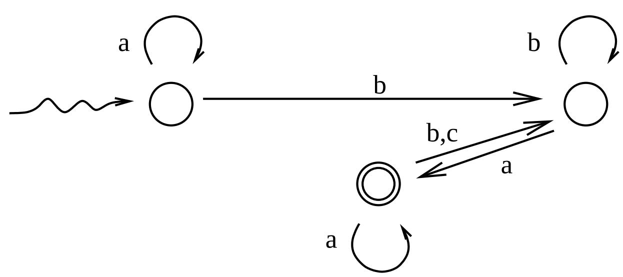

Figure 1: Minimal DFA equivalent to the NFA

68. Suppose a machine M_1 accepts a language L. Prove that there is some machine M_2 which enumerates L.