CP405 Theory of Computation
Final Test Practice Questions

1. Create a DFA for the language L = {w: w has exactly 3 a's} with Σ = {a, b}.

2. Create a DFA for the language described by the regex: ab*ab*.

3. Create an NFA for the language L = {w: w has the substring bab} with Σ = {a, b}.

4. Give a high-level description for a PDA that accepts the language L = {w#w^r: where w^r is the substring w in reverse order}.

5. Give a context-free grammar that produces the language in Question #4.

6. Give an implementation-level description of a TM that copies its input string and then halts.

7. Write out a truth table for a 2-input NAND gate.

8. How large is the powerset of the set A = {a, b, c, d}?

9. Why is the powerset used when defining transition functions for machines that have non-determinism?

10. Prove that if a language, L, is regular then so is LL (two strings in L concatenated together).

11. In high-level terms, describe how one could convert an arbitrary NFA into an equivalent DFA.

12. How do you prove a language is regular?

13. How do you prove a language is not regular?

14. Why do we choose a pumping length, p, that is the number of states in a finite automaton when proving the regular pumping lemma?

15. In the regular pumping lemma, how do we know that |xy| <= p?

16. Why do we often want to convert CFGs into Chomsky Normal Form?

17. What is the difference between context-free grammars and context-sensitive grammars?

18. For the proof of the context-free pumping lemma, how do we know

that |vy| >= 1?

19. Is it possible for a PDA to loop forever?

20. When does a PDA terminate its execution?

21. Can a TM have an infinite number of states in Q?

22. Can a TM have an infinite number of configurations?

23. Are TMs guaranteed to complete their executions in a finite amount of time?

24. For L = $\{a^n b^n c^n : n > 0\}$, give an implementation-level TM description of a TM that accepts L.

25. What is the pigeonhole principle? Where have we used it in this class so far?

26. Can non-deterministic TMs accept languages that deterministic TMs cannot?

27. If a language L is not regular, what can we say about |L|?

28. Prove the language L = $\{a^x b^y c^z : x * y = z\}$ is not regular.

29. With Σ = {0, 1}, prove the language L = {x#y: x is a substring of y} is not context-free. (Hint: consider w = $0^p 1^p \# 0^p 1^p$).

30. How are NFAs different from DFAs?

31. What does it mean for a language to be undecidable?

32. Give two examples of undecidable languages.

33. Write a brief outline of why A_{CFG} is decidable.

34. Using a 4-register counter machine with the instructions below, write an algorithm to calculate reg1 * reg2.

INC(x) add 1 to register x
DEC(x) subtract 1 from register x
CLR(x) set register x to 0
CPY(x, y) copy reg. x to reg. y
JNZ(x, z) if reg. x ≠ 0, jump to instruction z
JMP(z) jump immediately to instruction z
HALT terminate

35. What does it mean for a model/machine to be Turing-Complete?

36. Is it possible to write a Python function to analyze any Java program source code file and detect if the Java code will always produce the correct results?

37. What does ⟨M, w⟩ mean?

38. Is the infinite language aab*(bc)*a decidable?

39. What is the halting problem?

40. In the diagonalization proof of the uncountability of $\mathbb{R}$, what was the contradiction? What assumption must have been false?

41. Where is the contradiction in the diagonalization proof for the halting problem for TMs? What does it imply?

42. When discussing Rice's Theorem, we defined what it meant for TMs to have a certain property. Define what is meant by "property" in this context.

43. What contradiction was reached in the proof of Rice's Theorem? What assumption was false?

44. How many enumerable languages are there?

45. How many binary languages are there?

45. What is the difference between a decider and a recognizer?

46. What is the difference between a recognizer and an enumerator?

47. Name two languages that are decidable, but not enumerable, or explain why none can exist.

48. HALT is defined as {⟨M, w⟩: TM M halts on input w}. Is HALT enumerable? Why or why not?

49. How were binary sequences used in the proof that the number of binary languages is uncountable?

50. If a language and its complement are both enumerable, how do we know that language is decidable?

51. Name one language that is not enumerable, or explain why none can exist.

52. Is HALT enumerable?

53. In the proof for showing EQ_{TM} is not enumerable, what machine was constructed and what was the contradiction that was formed?

54. If a function is in the class $O(n^2)$, what does this mean mathematically?

55. Is linear search an element of P?

56. Is linear search an element of NP?

57. How do we define time complexity for TMs?

58. Given a 3SAT problem with 12 variables, how many possible solutions would a brute force algorithm have to check?

59. Define the language class NP.

60. The definition of NP states that the length of a problem's solution must be within a polynomial of the problem's description. Why is this important?

61. Are there problems in NP that are not in P?

62. What does it mean for a problem to be NP-Complete?

63. Give the mathematical definition for a polynomial-time reduction. Explain what it means in one sentence.

64. How can we show that a new problem is in the class NP?

65. How can we show that a new problem is in the class NP-Complete?

66. How do we define space complexity for TMs?

67. What is the space complexity of 3SAT?

68. What is the space complexity of linear search?

69. If someone found a polynomial-time solution to an NP-Complete problem, how quickly could we solve all problems in NP?

70. Does there exist a polynomial-time algorithm for solving subset sum problems?