University of Nevada, Las Vegas Computer Science 456/656 Spring 2024

Topics Covered on Examination February 7, 2024

1. Fundamentals.
   (a) What is an alphabet?
   (b) What is a symbol?
   (c) What is a language?
   (d) Decidable and undecidable languages. A language L is *decidable* (also called *recursive* if some machine decides L.
   (e) Computable and uncomputable functions. A function F is *computable* (also called *recursive* if some machine computes F.
   (f) Proof by contradiction. Assume a statement is true, then use logic to prove, from that assumption, that $2 + 2 = 3$, or some other statement known to be false. Then the original statement is false.
2. Finite state automata, and their relationship with regular languages.
   (a) Draw a state diagram.
   (b) Minimize a DFA.
   (c) Find a DFA equivalent to an NFA.
   (d) What language does this DFA, or NFA, accept?
   (e) T/F or fill-in-the blank questions.
3. Regular expressions.
   (a) Find a regular expression.
   (b) The language described by a regular expression. Find, in the list given below, the regular expression for each language.
   (c) T/F or fill-in-the blank questions.
4. State the pumping lemma accurately. The quantifiers must be properly expressed. If you have all the right words in some wrong order you might get no credit.
5. Regular grammars.
   (a) Left-regular (left-linear) or right-regular (right-linear).
   (b) A grammar that generates the language accepted by an NFA.
   (c) T/F or fill-in-the blank questions.
6. Other questions about regular languages.
   (a) The empty string.
   (b) Kleene closure.
   (c) Closure properties of the class of regular languages.
   (d) Homomorphism.

(e) Every finite language is regular.

(f) Regular sets of numbers.

- Is the set of terms of an arithmetic sequence regular?
- Is the set of prime numbers regular?

(g) T/F or fill-in-the blank questions.

7. Context-free grammars and languages.

(a) Derivations.

i. Derivation of a string.

ii. Left-most and right-most derivations.

iii. Parse trees.

iv. Ambiguity and uniqueness of derivations.

v. T/F or fill-in-the blank questions.

(b) Equivalence of grammars.

i. The CF grammar equivalence problem is undecidable.

(c) Chomsky Normal Form.

i. The CYK algorithm.

(d) Push-down automata. (PDAs)

i. Diagram of a PDA

ii. Deterministic push-down automata. (DPDAs)

- What does that mean?
- End-of-file symbol.

iii. How does a PDA accept a string?

- Input file must be empty.
- Empty stack.
- Final state.
- Empty stack and final state.

iv. T/F or fill-in-the blank questions.

8. Logic

(a) Truth tables.

(b) Universal and existential quantifiers.

9. Computational Complexity

(a) Polynomial functions.

(b) Language classes.

i. The language class $\mathcal{P}$-TIME, usually just called $\mathcal{P}$.

ii. The language class $\mathcal{NP}$. Two definitions.

- Polynomially many steps of a non-deterministic machine.
- Deterministic verification in polynomial time, if the answer is **true**.

iii. $\mathcal{NP}$-completeness.

iv. Some $\mathcal{NP}$-complete languages/ problems:

- SAT.
- 3-SAT.

CSC 456/656 Fall 2023 First Examination Problems to Study

1. True or False. 5 points each. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

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

(ii) **T** The class of regular languages is closed under intersection.

(iii) **O** $\mathcal{P}$–TIME = $\mathcal{NP}$.

(iv) **T** The class of regular languages is closed under Kleene closure.

(v) **T** The class of context-free languages is closed under union.

(vi) **F** The class of context-free languages is closed under intersection.

(vii) **F** The set of binary numerals for prime numbers is a regular language.

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

(ix) **F** The complement of any context-free language is context-free.

(x) **T** The complement of any recursive (that is, decidable) language is recursive.

(xi) **T** If Σ is an alphabet, then Σ^* is a regular language.

(xii) **F** If L is a language and L^* is a regular language, then L must be a regular language. **(Think!)**

(xiii) **T** The class of languges which are **not regular** is closed under intersection. **(Think!)**

(xiv) **F** A minimal DFA equivalent to an NFA with n states must have 2^n states.

(xv) **O** If a non-derministic machine can solve a given problem in polynomial time, then there is a deterministic machine which can solve the same problem in polynomial time.

(xvi) **T** If a non-derministic machine can solve a given problem in polynomial time, then there is a deterministic machine which can solve the same problem in exponential time.

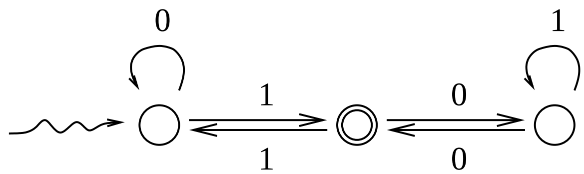
2. Give an example of a language which is context-free but not regular.

   $\{a^n b^n\}$

3. Give an example of a language which is not context-free.

   $\{a^n b^n c^n\}$

4. Let L be the language of all binary strings encoding numbers which are equivalent to 1 modulo 3, where leading zeros are allowed. Thus, $L = \{1, 01, 001, 100, 111, 0100, 0111, 1010, \ldots\}$. Draw a DFA which accepts L. (You need only three states.)



5. Let G be the CF grammar given below, where E is the start symbol. Show that G is ambiguous by giving two different **rightmost** derivations for the string $x - y * z$.

   1. $E \to E - E$
   2. $E \to E * E$
   3. $E \to x$
   4. $E \to y$
   5. $E \to z$

   $E \Rightarrow E - E \Rightarrow x - E \Rightarrow x - E * E \Rightarrow x - y * E \Rightarrow x - y * z$

   $E \Rightarrow E * E \Rightarrow E - E * E \Rightarrow x - E * E \Rightarrow x - y * E \Rightarrow x - y * z$

   The first derivation is the one that respects the usual precedence of the operators.

6. Give a grammar for the language accepted by the NFA shown in Figure 1 below.

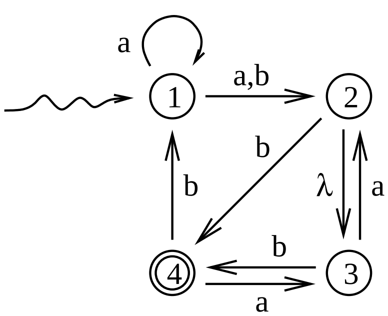
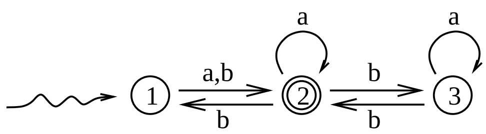


Figure 1: NFA for problems 6 and 9.

$S \to aS$
$S \to aA$
$S \to bA$
$A \to B$
$A \to bC$
$B \to bC$
$B \to aA$
$C \to aB$
$C \to bS$
$C \to \lambda$

7. Write a regular expression for the language accepted by the following NFA



$(a + b)(a + b(a + b) + ba * b)*$

8. Write the pumping lemma for regular languages *correctly.* Pay close attention to the order in which you write the quantifiers. If you have all the correct words in the wrong order, you still might get no credit. **This will not be on the September 27 examination.**

For any regular language L, there is an integer p (which is called the pumping length of L) such that

For any $w \in L$ such that $|w| \geq p$

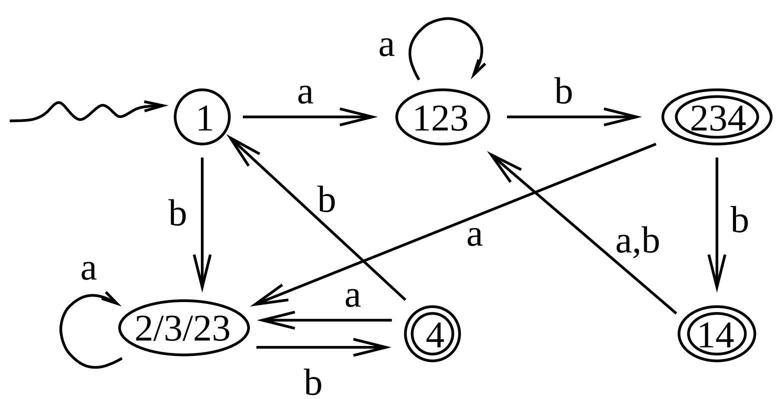
There exist strings x, y, z such that the following four statements hold:

1. $w = xyz$
2. $|barredxy \leq p$
3. $|barredy \geq 1$
4. for any integer $i \geq 0$, $xy^iz \in L$

9. Draw a minimal DFA equivalent to the NFA shown in Figure 1 in problem 6 above. Show the transition table, and also show the matrix used for minimizing the DFA.

	a	b
1	123	23
2	23	4
3	23	4
((4	3	1
123	123	234
23	23	4
((234	23	14
((14	123	234

	1	2	3	4	123	23	234	14
1	O	X	X	X	X	X	X	X
2	X	O	O	X	X	O	X	X
3	X	O	O	X	X	O	X	X
4	X	X	X	O	X	X	X	X
123	X	X	X	X	O	X	X	X
23	X	O	O	X	X	O	X	X
234	X	X	X	X	X	X	O	X
14	X	X	X	X	X	X	X	O



University of Nevada, Las Vegas Computer Science 456/656 Spring 2022

Topics Covered on Examination February 23, 2022

1. Fundamentals.
   (a) What is an alphabet?
   (b) What is a symbol?
   (c) What is a language?
   (d) Decidable and undecidable languages.
   (e) Computable and uncomputable functions.
   (f) Proof by contradiction.
2. Finite state automata, and their relationship with regular languages.
   (a) Draw a state diagram.
   (b) Minimize a DFA.
   (c) Find a DFA equivalent to an NFA.
   (d) What language does this DFA, or NFA, accept?
   (e) T/F or fill-in-the blank questions.
3. Regular expressions.
   (a) Find a regular expression.
   (b) The language described by a regular expression.
   (c) T/F or fill-in-the blank questions.
4. State the pumping lemma accurately. The quantifiers must be properly expressed. If you have all the right words in some wrong order you might get no credit.
5. Regular grammars.
   (a) Left-regular (left-linear) or right-regular (right-linear).
   (b) A grammar that generates the language accepted by an NFA.
   (c) T/F or fill-in-the blank questions.
6. Other questions about regular languages.
   (a) The empty string.
   (b) Kleene closure.
   (c) Closure properties of the class of regular languages.
   (d) Homomorphism.
   (e) Every finite language is regular.
   (f) Regular sets of numbers.
      - Is the set of terms of an arithmetic sequence regular?

- Is the set of prime numbers regular?

(g) T/F or fill-in-the blank questions.

7. Context-free grammars and languages.

   (a) Derivations.

      i. Derivation of a string.
      ii. Left-most and right-most derivations.
      iii. Parse trees.
      iv. Ambiguity and uniqueness of derivations.
      v. T/F or fill-in-the blank questions.

   (b) Equivalence of grammars.

      i. The CF grammar equivalence problem is undecidable.

   (c) Chomsky Normal Form.

      i. The CYK algorithm.

   (d) Push-down automata. (PDAs)

      i. Diagram of a PDA
      ii. Deterministic push-down automata. (DPDAs)
         - What does that mean?
         - End-of-file symbol.
      iii. How does a PDA accept a string?
         - Input file must be empty.
         - Empty stack.
         - Final state.
         - Empty stack and final state.
      iv. T/F or fill-in-the blank questions.

8. Logic

   (a) Truth tables.
   (b) Universal and existential quantifiers.

9. Computational Complexity

   (a) Polynomial functions.
   (b) Language classes.

      i. The language class $\mathcal{P}$-TIME, usually just called $\mathcal{P}$.
      ii. The language class $\mathcal{NP}$. Two definitions.
         - Polynomially many steps of a non-deterministic machine.
         - Deterministic verification in polynomial time, if the answer is **true**.
      iii. $\mathcal{NP}$-completeness.
      iv. Some $\mathcal{NP}$-complete languages/ problems:
         - SAT.
         - 3-SAT.