CHAPTER

9 Review

9.1 Definitions in Chapter 9

9.1 Information

9.3 Finite Automaton

9.4 String

9.5 Finite-State Machine with Output

9.6 Symbols; Alphabet; Word; Null String

9.7 Language

9.8 Regular Grammar

9.9 Derivation

9.10 Language Generated by a Grammar

9.11 An Informal Definition of Regular Expressions

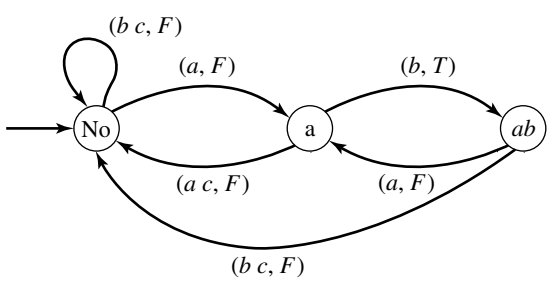
9.12 A Formal Definition of Regular Expressions

9.13 Nondeterministic Finite Automaton

9.21 The Complement of aLanguage

9.2 Sample Exam Questions

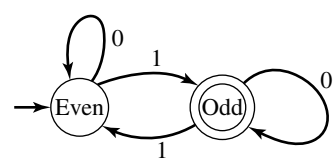
1. Reproduce the diagram of Shannon's model for a communication system. Label the diagram.
2. A collection of four messages, $S = \{m_1, m_2, m_3, m_4\}$, has respective probabilities, $\{\frac{1}{2}, \frac{1}{8}, \frac{1}{4}, \frac{1}{8}\}$. What is the average information in a randomly received message from this set?
3. Create a finite automaton that recognizes any input string that contains exactly two a's. Assume the input alphabet is $\{a, b, c\}$.
4. What is the output string that results when the input string *abbcbab* is sent to the following finite-state machine? What is a likely purpose for this machine?



5. Describe the productions in a regular grammar.
6. Let Σ be a set of symbols. Define the Kleene closure of Σ.
7. Create a single regular expression that matches any name in one of the following forms (case is important). Assume that both first and last names contain two or more characters. Use ␣ to represent the space character. Do not use any Perl extensions.

   Mr. First Last Mrs. First Last Ms. First Last First Last

   Note that "mr. bob Smith" should *not* be matched, since the "m" in "mr." and the "b" in "bob" are lowercase.
8. Produce a regular grammar, $\mathcal{G}$, such that $L(\mathcal{G})$ is the set of input strings recognized by the following finite automaton. The set of input symbols is {0, 1}.



9. Match each type of grammar with a state model that is capable of recognizing strings in the language generated by the grammar.

   Grammars Context free; regular; context sensitive; phrase structured

   State machines Pushdown automaton; finite automaton, Turing machine
10. State the Church–Turing thesis.

9.3 Projects

Mathematics

1. Write a brief expository paper that introduces Shannon's theory of channel capacity.
2. Master the proof of Theorem 9.14. Then write a clear presentation of the proof together with any necessary background information.
3. Write a brief expository paper on the Church–Turing thesis.
4. Write a brief expository paper about primitive recursive functions and computable functions.

Computer Science

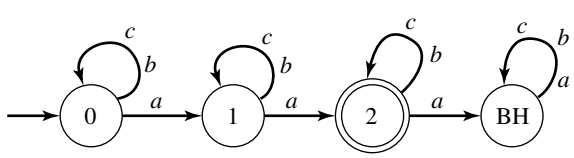
1. Write a GUI-based program that simulates finite automata and/or finite-state machines with output.
2. Write a brief introductory paper about the use of formal languages in the design of compilers.
3. Write a brief expository paper about LR grammars. Include a discussion about the meaning of "LR."
4. Write a detailed report about Turing machines. Include a simple program that will run on a Turing machine.

9.4 Solutions to Sample Exam Questions

1. See Figure 9.1 on page 534.

2. $$\begin{aligned} I(S) &= -\sum_{k=1}^{4} p_k \log_2(p_k) \\ &= -\left(\frac{1}{2}\cdot\log_2\left(\frac{1}{2}\right) + \frac{1}{8}\cdot\log_2\left(\frac{1}{8}\right)\right. \\ &\quad \left. + \frac{1}{4}\cdot\log_2\left(\frac{1}{4}\right) + \frac{1}{8}\cdot\log_2\left(\frac{1}{8}\right)\right) \\ &= -\left(-\frac{1}{2} - 3\cdot\frac{1}{8} - 2\cdot\frac{1}{4} - 3\cdot\frac{1}{8}\right) \\ &= \frac{7}{4} \end{aligned}$$

3. The following state diagram shows such a finite automaton.



4. The following table summarizes the output string (in the final row), as well as the sequence of states.

Input	a	b	b	c	b	a	b
State	a	ab	No	No	No	a	ab
Output	F	T	F	F	F	F	T

This finite-state machine outputs a T if the previous two input characters are a and then b. It outputs an F in all other cases. So its purpose is to detect the substring ab.

5. The productions in a regular grammar are all in the form $N \to \nu$, where N is a nonterminal and $\nu \in (\Sigma \cup \Delta)^*$. The string ν satisfies the following conditions:
   - It must contain at most one nonterminal symbol.
   - If ν contains a nonterminal, it must be the rightmost symbol.
   - If the only terminal symbol is λ, then there can be no nonterminal symbol.
   - The string ν must contain at least one terminal symbol.

6. The Kleene closure of Σ is denoted by Σ^* and defined as
$$\Sigma^* = \{\lambda\} \cup \Sigma \cup \Sigma^2 \cup \Sigma^3 \cup \cdots = \cup_{i=0}^{\infty} \Sigma^i.$$
Intuitively, Σ^* is the collection of all finite length strings over Σ (including the empty string).

7. The beginning and end of line metacharacters help avoid surprises such as matching "mr. Jim Smith" (since the subpattern "Jim Smith" is a legal match).

```
^((Mr|Mrs|Ms)[.]␣)?[A-Z][a-z]+␣ [A-
Z][a-z]+$
```

8. Follow the proof of Theorem 9.15.

   The set, Σ, of terminal symbols will be the input symbols of the finite automaton. Thus, $\Sigma = \{0, 1\}$. The start state is Even. The set of nonterminals, Δ, corresponds to the states in the finite automaton. Thus, $\Delta = \{\text{Even}, \text{Odd}\}$. The productions are determined by the transitions. The standard productions (of the form $N \to aM$) are

   Even $\to$ 0 Even
   Even $\to$ 1 Odd
   Odd $\to$ 0 Odd
   Odd $\to$ 1 Even

   The productions that arise from transitions to final states are

   Even $\to$ 1
   Odd $\to$ 0

   To raise confidence that this grammar works, try a few examples. Here is a derivation of 1101:

   $$\text{Even} \Rightarrow 1\ \text{Odd} \Rightarrow 11\ \text{Even} \Rightarrow 110\ \text{Even} \Rightarrow 1101$$

   Notice that 101 cannot be derived. An attempted derivation would look like

   $$\text{Even} \Rightarrow 1\ \text{Odd} \Rightarrow 10\ \text{Odd} \Rightarrow ???.$$

   The only choice for continuing with an input of 1 and Odd as the only nonterminal is Odd $\to$ 1 Even. Since there is no production of the form Even $\to \lambda$, there is no way to stop with just 101. Thus, 101 is not in the language generated by this grammar. (It shouldn't be, since the finite automaton recognizes binary strings with an odd number of 1s.)

9. See Table 9.9 on page 584.

10. **Church–Turing thesis:** A Turing machine is capable of performing any computable algorithm.