Summary of Chapter 1

An **alphabet** is a set of symbols.

A **string** is a finite sequence of symbols drawn from some alphabet.

A **language** is any set of strings.

The **empty string** is denoted ε.

A **finite automaton** (FA) is a device that recognizes a language.

An FA has finite memory and an input tape: each input symbol that is read causes the machine to update its state based on its current state and the symbol read.

An FA **accepts** the input if it is in an accept state at the end of the string; otherwise the input is rejected.

Summary of Chapter 2

A **regular expression** (RE) is built up from individual symbols using the three Kleene operators.

The Kleene operators are **union** ($+$), **concatenation**, and star (*).

The **star** of a language is obtained by all possible ways of concatenating strings of the language, repeats allowed.

The empty string is always in the star of a language.

A **regular language** is one that has an FA or an RE.

Summary of Chapter 3

A **nondeterministic finite automaton** (NFA) can have zero, one, or multiple transitions corresponding to a particular symbol.

An NFA is defined to accept the input *if there exists some choice of transitions* that cause the machine to end up in an accept state.

Nondeterminism can also be viewed as a tree, or as a "guess-and-verify" concept.

One can also have **ε-transitions**, where the NFA can change state without consuming an input symbol.

Kleene's theorem says that the following are equivalent for a language:
there is an FA for it;
there is an NFA for it; and
there is an RE for it.

The proof of Kleene's algorithm provides an algorithm to convert from one form to another; the conversion from NFA to DFA is the **subset construction**.

Summary of Chapter 4

Regular languages are **closed** under union, concatenation, star, and complementation.

To show that a language is nonregular, one can show that there is an *infinite set of pairwise distinguishable strings*.

To show that a language is nonregular, one can use the **Pumping Lemma** and show that there is some string that cannot be pumped.

Summary of Chapter 5

Applications of finite automata include string matching algorithms, network protocols and lexical analyzers.