

## Chapter 1: Introduction to Formal Languages

A formal language is a set of strings formed from a specific alphabet and rules. These are used in computer science, especially in compiler design and automata theory.

Examples of alphabets:  $\{a, b\}$ ,  $\{0, 1\}$

A string is a sequence of characters. For example, 'ab', 'aaab', '01', etc.

Exercises:

1. What is a formal language?
2. Give 3 examples of strings over the alphabet  $\{0, 1\}$ .
3. Is 'abc' a valid string over the alphabet  $\{a, b\}$ ? Why or why not?

Test:

1. Define 'alphabet' and 'string'.
2. Explain what makes a set of strings a language.

## Chapter 2: Finite Automata

Finite automata are machines used to recognize patterns within input strings. There are two types: deterministic (DFA) and non-deterministic (NFA).

A DFA has:

- A finite set of states

## Formal Languages: Simplified English Edition

- A start state
- A set of accepting states
- A transition function

### Exercises:

1. Draw a DFA that accepts strings ending with 'ab'.
2. What are the key differences between DFA and NFA?

### Test:

1. Explain the structure of a DFA.
2. True/False: An NFA can have multiple transitions for the same input from a state.