Title Slide - **Title:** Theory of Computation - **Subtitle:** Foundations of Computer Science - **Presented by:** [Your Name] - **Image:** Abstract image representing logic gates or binary codes

---

Slide 1: Introduction - **What is Theory of Computation?** - Study of abstract machines and the problems they can solve - Foundation for computer science and programming - **Main Areas:** - Automata Theory - Formal Languages - Computability Theory - Complexity Theory - **Image:** Conceptual diagram of computation branches

---

Slide 2: Automata Theory - **Definition:** Study of abstract machines and problems they solve - **Types of Automata:** - Finite Automata (DFA, NFA) - Pushdown Automata (PDA) - Turing Machines (TM) - **Diagram:** Example of a DFA with states and transitions

---

Slide 3: Finite Automata - **Deterministic Finite Automata (DFA):** - Defined by 5-tuple (Q, Σ, δ, q0, F) - One transition per symbol from each state - **Non-Deterministic Finite Automata (NFA):** - Multiple transitions allowed - Epsilon (ε) transitions possible - **Image:** Comparison diagram of DFA and NFA

Slide 4: Regular Languages - **Definition:** Languages accepted by finite automata - **Representations:** - Regular expressions - Regular grammars - **Closure Properties:** Union, Concatenation, Kleene star - **Image:** Venn diagram of regular languages and operations

---

Slide 5: Pushdown Automata (PDA) - **Definition:** Finite automaton with a stack - **Used for:** Context-Free Languages - **Components:** Stack, states, input symbols, transition function - **Diagram:** Example of PDA accepting language a^n b^n

---

Slide 6: Context-Free Grammars (CFG) - **Definition:** Grammar generating context-free languages - **Structure:** Consists of variables, terminals, start symbol, productions - **Application:** Parsing in compilers - **Image:** Parse tree for arithmetic expression

---

Slide 7: Turing Machines - **Definition:** Abstract model of computation - **Components:** Tape, head, state register, transition function - **Purpose:** Model general computation - **Diagram:** Turing machine reading and writing on tape

---

Slide 8: Computability Theory - **Key Concepts:** - Decidable and undecidable problems - Halting problem - **Church-Turing Thesis:** Anything computable can be computed by a TM - **Image:** Chart showing decidable vs. undecidable problems

---

Slide 9: Complexity Theory - **Purpose:** Classify problems based on resource usage - **Classes:** - P: Polynomial time - NP: Non-deterministic Polynomial time - NP-complete, NP-hard - **Diagram:** Venn diagram showing P, NP, NP-complete

---

Slide 10: Applications of Theory of Computation - **Compiler Design** - **Artificial Intelligence** - **Cryptography** - **Software Engineering** - **Image:** Icons or illustrations representing each field

---

Slide 11: Conclusion - **Recap:** - Core areas: Automata, Grammars, Turing Machines - Real-world relevance - **Encouragement:** Explore advanced topics and applications - **Thank You Slide** - **Image:** Inspirational quote with computational background