Theory Of Computation

Automata: FSM, Inputs, Outputs, DFA, NFA, Computability: Turing Machines, Complexity: BigO

An automaton (plural: automata) is a logical model of a machine that, based on input events, transitions from state to state. To describe an automaton we need to identify its states, the set of acceptable inputs and the expected outputs, and describe how transitions work. There are a number of optional additional features to constructing automata and these can add a range of extra capabilities.

Automata theory is used throughout mathematics and programming (e.g. compilers, protocols). It's sometimes so natural that users are often unaware of its presence.

Computability theory shows how an abstract machine can be subjected to mathematical reasoning and certain important characteristics can be reliably proven. We explore the best known of these – a Turing Machine. Complexity theory helps classify the degree of difficulty (from trivial to impossible) there is in solving a given computational problem.

A clear understanding of the theory of computation helps everyone on a team have a richer appreciation of how automata, computability theory and complexity theory can be beneficial to a product's architecture.

Target Audience

This course is aimed mathematicians and software developers who wish to become familiar with important aspects of how mathematics plays a foundational role in computation.

Prerequisites

Attendees need a good understanding of mathematics and software programming.

Contents of One-Day Training Course

Overview Of Automata Theory

Practical uses of automata
Overview of automata theory
Deterministic vs. non-deterministic
How different automata vary

Types of Automata

In increasing order of complexity:
* Finite state machine
* Pushdown automata
* Linear bounded automata
* Turing machine
What more complex automata brings

What is Needed to Build

States
Inputs – what drives transitions
Outputs – result of transitions
Transitions

States And Transitions

Identifying states
Optionally - identifying initial / final states
May be more than one
Transition function

Deterministic Automaton

A given sequence of inputs will result in a given set of state transitions
A set of states
A set of inputs
the next state function
the final predicate

Non-Deterministic Automaton

Impact of non-determinism
Transition relation
Converting a NFA to a DFA

Specialist Automata Topics

Acceptance conditions
Automata with an infinite number of states
Cooperation between multiple automata
Relationship to computational theory
Asynchronicity

Computability Theory

What is computability & recursion theory?
Model of computation - mathematically describing computation
Examining the properties of computation
Reverse mathematics

Deep Dive: Turing Machine

What is a Turing Machine?
How does it work?
What does its operation demonstrate?
Understanding this abstract machine brings many benefits

Intro to Complexity Theory

Computation with large numbers of steps and states can have performance issues
Specifically for these, need to consider variation of approaches and how to measure complexity

Advanced Complexity Theory

Trying to estimate amount of resources needed for particular compute workload
Exploring the limits of computation
Can a problem be solved at all?

Project

Use of theory of computation in a non-trivial project to show its benefits in a practical setting