Automata Theory

Finite State Machines, State, Inputs, Outputs, DFA, NFA, Turing Machines, Uses

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.

The abstract machine represented by an automaton can be subjected to mathematical reasoning and certain important characteristics can be reliably proven.

Automata theory is used throughout mathematics and programming. It is sometimes so natural that users are often unaware of its presence. We see it as a substrate feature of lexical analysis / parsing in compilation, as an alternative to the tableau algorithm in description logic, as finite state machines in any server product, as the construct used by generated networking comms layer from linear logic/session types & neural networks in AI.

Having a clear understanding of automata theory helps everyone on a team have a richer appreciation of how automata can be beneficial to a product's architecture.

Target Audience

This course is aimed mathematicians and developers who wish to become familiar with the theoretical and practical usage of automata theory.

Prerequisites

Attendees need a good foundation in mathematics and 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

Automata and Category Theory

Representing automata as categories
Morphisms for automata
The Aut Category

Use Of Automata In Compilers

Modern lexical analysis and parsing are essentially the use of automata with some supporting code (to load source file, or to construct abstract syntax tree)
Dotted items to walk through input stream
Base items (shift/reduce) & non-base items

A Stack For Automata

Sometimes it can be interesting to allow states to have memory
A stack with push and pop operations
Use of such a stack for compilers
Designing pushdown automata engines

Complexity And Automata

Automata with large numbers of states can have performance issues
Specifically for these, need to consider variation of approaches (pruning, compression, combination, ..)

Project

Use of automata theory in a non-trivial project to shows its benefits in a practical setting