Finite-State Machines (Automata) lecture 12

- a simple form of computation
- used widely
- one way to find patterns
How can we use logic to specify the transitions?

This is a **non-deterministic** system. We define a next state **relation**.
ATM

- Insert card
- PIN ok
- Balance
- Withdraw
- Amount
- Choose a/c
- Yes
- No
- Take money
- Return card
- Wrong PIN
- Return card
- £
Counting trains

A axle sensor (detects passing wheels)

from-a-to-b: a↓; b↓; a↑; b↑

from-b-to-a: b↓; a↓; b↑; a↑
Finite-state machines

inputs: \(a^\uparrow, a^\downarrow, b^\uparrow, b^\downarrow\)

outputs: from-a-to-b, from-b-to-a
Hierarchical FSMs

carriage counter

inputs:
a2b, b2a

outputs:
A2B, B2A

a2b = from-a-to-b
Application Fields

Industry
  • real-time control, vending machines, cash dispensers, etc.

Electronic circuits
  • data path / control path
  • memory / cache handling
  • protocols, USB, etc.

Communication protocols
  • initiation and maintenance of communication links
  • error detection and handling, packet retransmission

Language analysis
  • natural languages
  • programming languages
  • search engines
A Decimal Number

\[ \text{alternative paths} \]

\[ \text{repetition} \]

\[ \text{skip} \]