University of Chinese Academy of Sciences

Logic Thinking

Automata and Turing Machines

zxu@ict.ac.cn
zhangjialin@ict.ac.cn

Outline

- Foundation of logic
  - Propositional Logic
  - Predicative Logic
- Automata and Turing Machines
- Power and Limitation of Computing
  - Mechanical Theorem Proving
  - Church-Turing Hypothesis

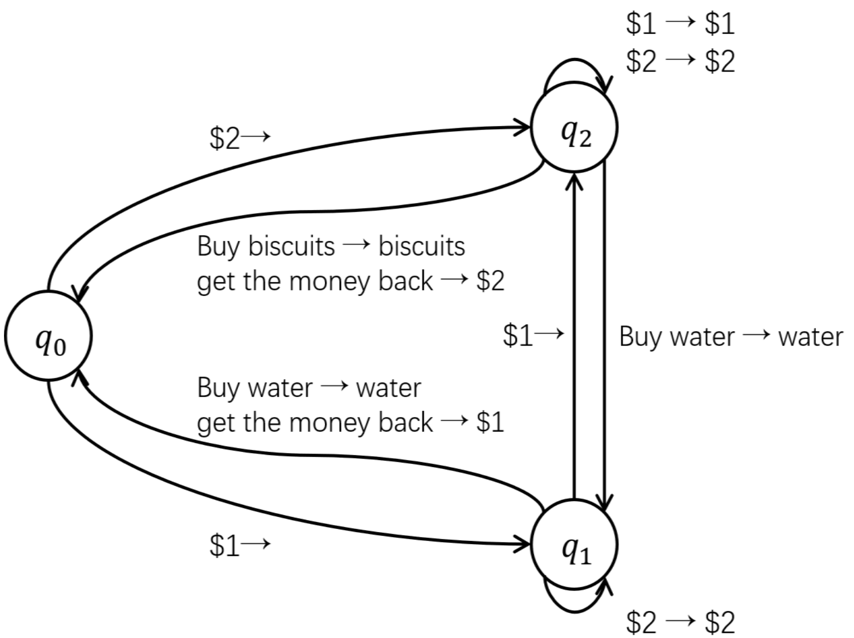
1.3 AUTOMATA AND TURING MACHINES

Automata

- Many computational processes are described using states
- A step in such a computational process involves computing the next state and/or the outputs from the current state and the inputs
- We study two types of automata
  - A finite state automaton has a finite state transition table
  - A Turing machine (TM) is an automaton, which has
    (1) a finite state transition table, and
    (2) an infinite tape
- Computer analogy
  - finite state automaton ~ CPU; TM ~ CPU+Memory+I/O

Example of a vending machine

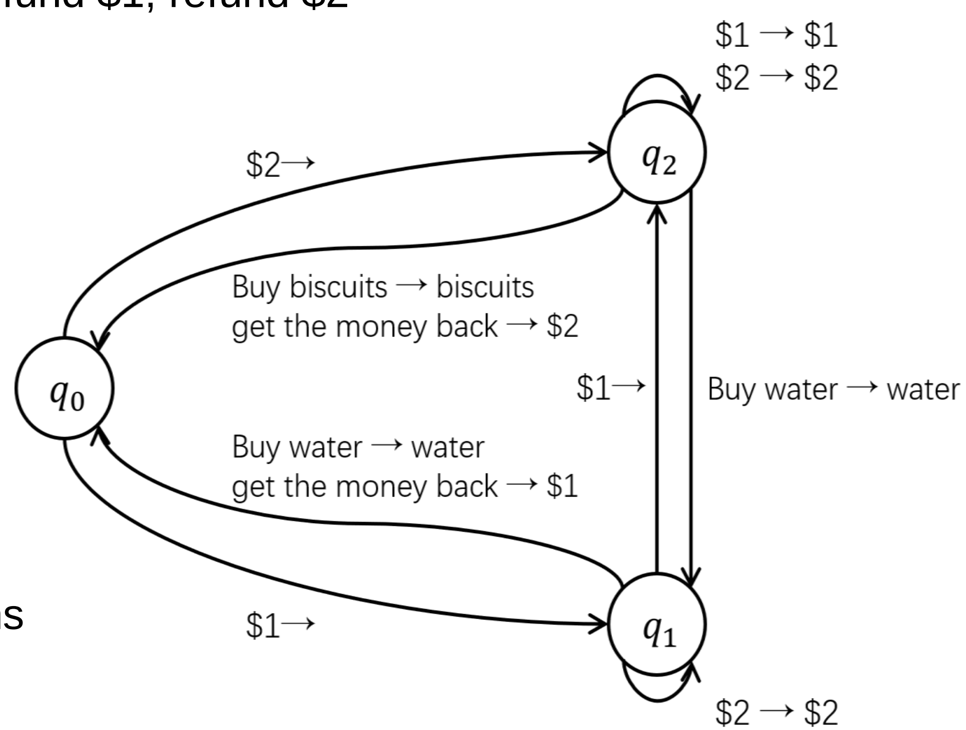
- This is a deterministic finite state automaton (DFA)
  - Can be specified by its state-transition diagram
- The machine has 3 states $q_0,\ q_1,\ q_2$ (i.e., nodes)
  - The initial state is q_0
- The buyer can perform 5 actions
  - Insert a $1 banknote
  - Insert a $2 banknote
  - Press "Buy water" button
  - Press "Buy biscuits" button
  - Press "Get money back" button
- Understanding transitions (labeled edges)
  - “$2→” in state q_0
  - “Buy water → water” in state q_2
  - “$1→$1”, “$2→$2” in state q_2



State-Transition Diagram

How to decide the state-transition diagram?

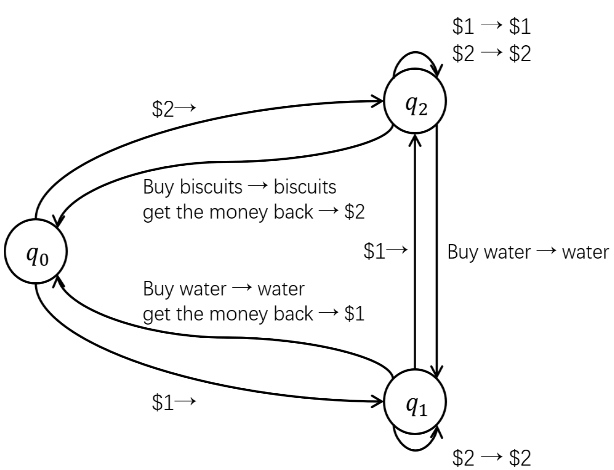
- This is art, often done by humans
- Some tips
  - First determine the inputs and outputs
    - 5 buyer input actions: insert $1 or $2; press 3 buttons
    - Machine outputs: water; biscuits; refund $1; refund $2
  - Then decide the states
    - The initial state: q_0
    - Already inserted $1: q_1
    - Already inserted $2: q_2
      - Insert a $2 bill, or
      - Twice insert a $1 bill
  - Then decide the transitions
    - For each state, there are 5 inputs
      - Thus, there are 5 transitions
    - In total, there are 3*5=15 transitions



  - The state transition diagram is incomplete!

A state-transition diagram can be rewritten as a state-transition table

- The state-transition table is complete, showing 15 transitions
- The state-transition diagram is incomplete
  - Showing only 11 transitions
  - Missing the 4 transitions in red



The initial state: q_0
Already inserted $1: q_1
Already inserted $2: q_2

Current State	Input	Output	Next State
q_0	Insert $1	Null	q_1
	Insert $2	Null	q_2
	Buy water	Null	q_0
	Buy biscuits	Null	q_0
	Get money back	Null	q_0
q_1	Insert $1	Null	q_2
	Insert $2	Output $2	q_1
	Buy water	Output water	q_0
	Buy biscuits	Null	q_1
	Get money back	Output $1	q_0
q_2	Insert $1	Output $1	q_2
	Insert $2	Output $2	q_2
	Buy water	Output water	q_1
	Buy biscuits	Output biscuits	q_0
	Get money back	Output $2	q_0

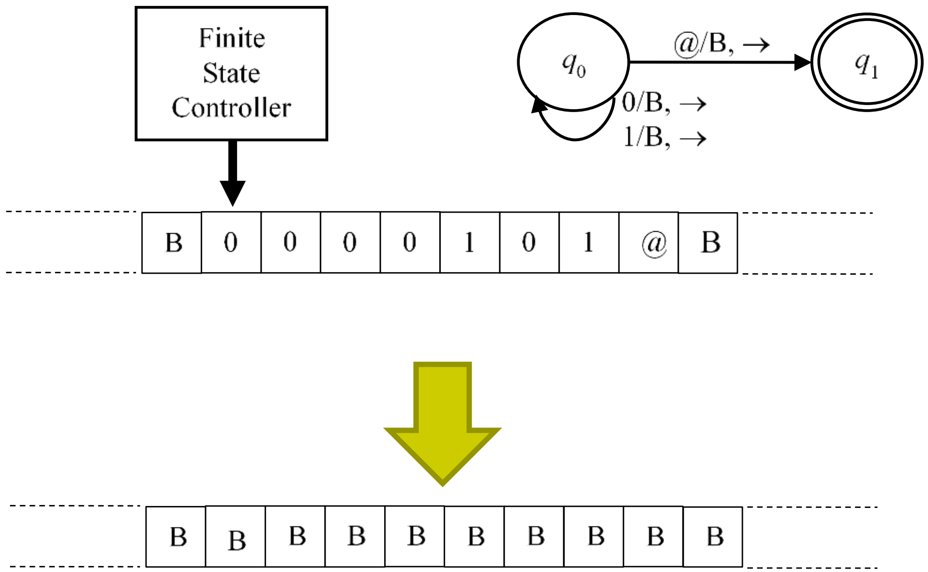
Some problems (and computational processes) cannot be modeled by DFA

- Finite-state automata cannot handle infinite states
  - A computational process involving potentially infinite number of states cannot be modeled by a finite-state automaton
- A simple example: recognizing palindromes
  - A palindrome is a character string that is the same when reading backwards
  - E.g., 1991 is a palindrome, so is 0100110001110000111100001110001100010
  - 1992 is not a palindrome
- Exercise***: show that palindromes cannot be recognized by finite automata

- Turing machines come to the rescue

Turing machine example

- In addition to the state-transition diagram stored in the Finite State Controller, there is an infinite tape in a Turing machine
- Cleanup: Replace every symbol by a blank (B)
  - The input data string is initially placed in the tape between two blanks

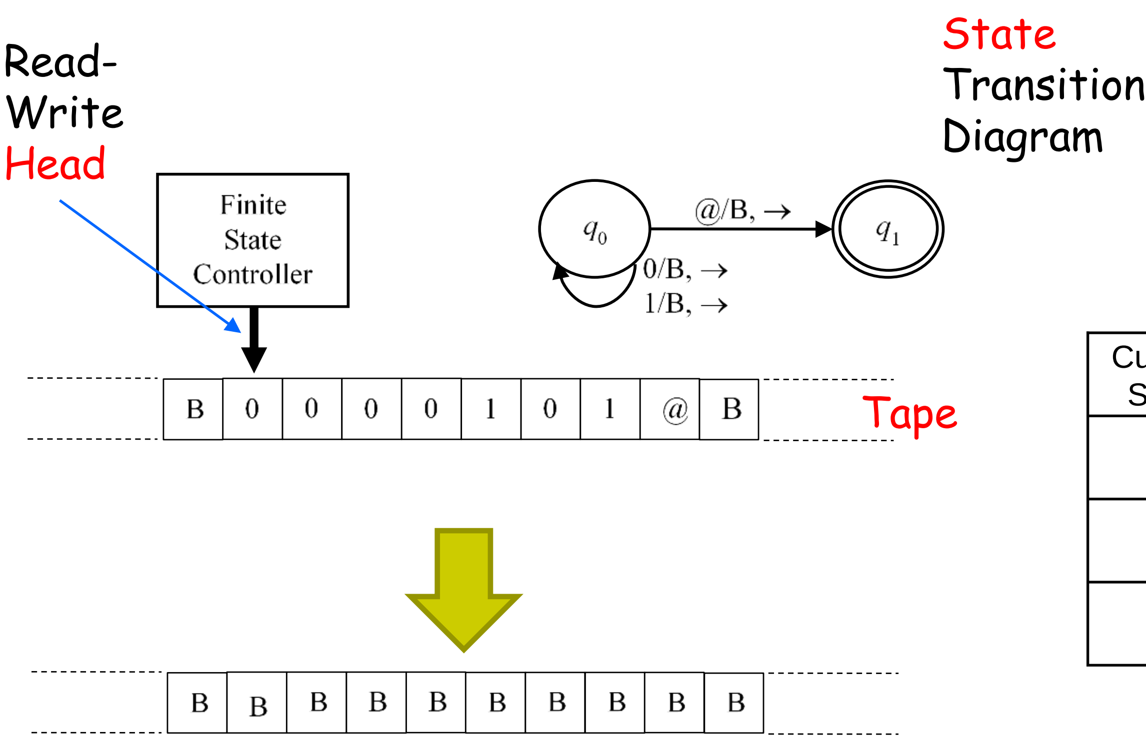


State-Transition Table

Current State	Symbol Read	Symbol to Write	Head Move	Next State
q_0	0	B	→	q_0
q_0	1	B	→	q_0
q_0	@	B	→	q_1 (Halt)

Turing machine example

- In addition to the state-transition diagram stored in the Finite State Controller, there is an infinite tape in a Turing machine
- Cleanup: Replace every symbol by a blank (B)
  - The input data string is initially placed in the tape between two blanks

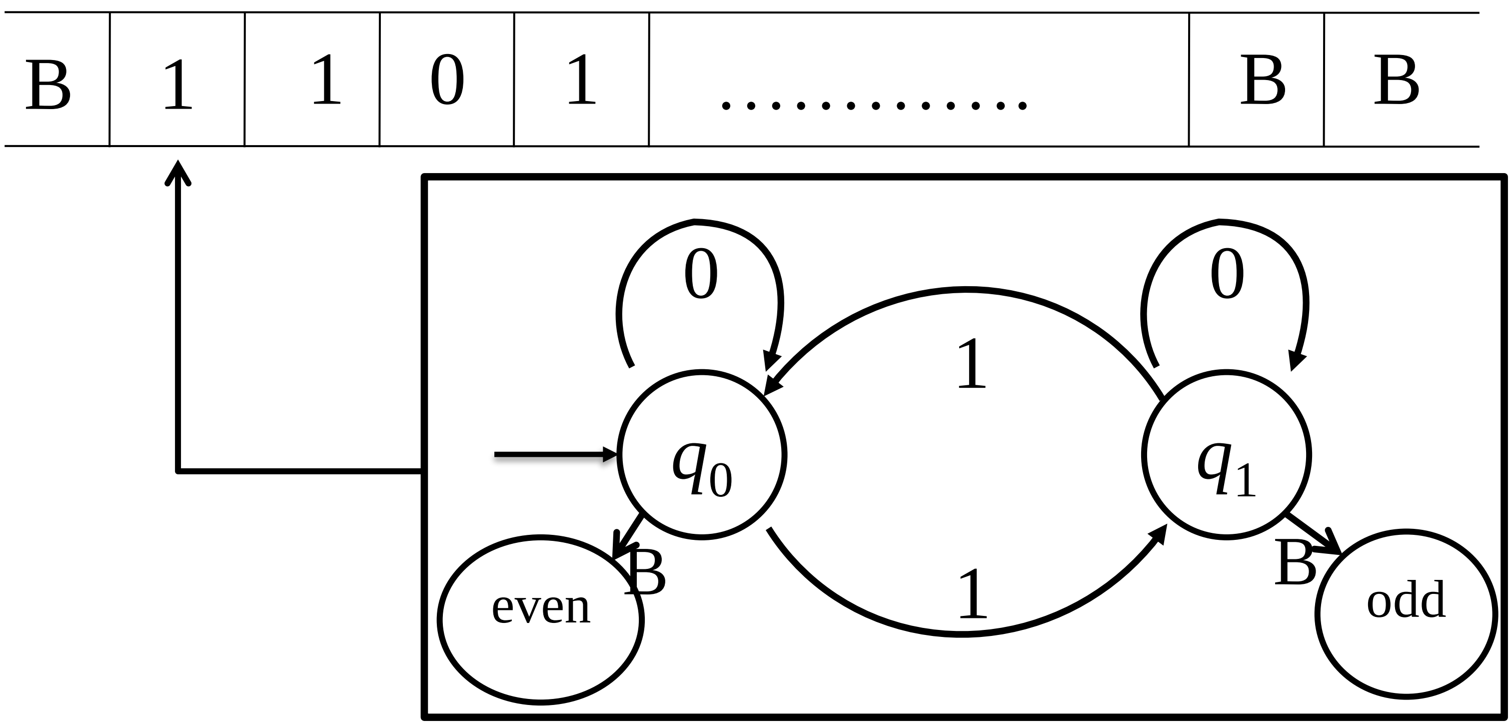


State-Transition Table

Current State	Symbol Read	Symbol to Write	Head Move	Next State
q_0	0	B	→	q_0
q_0	1	B	→	q_0
q_0	@	B	→	q_1 (Halt)

Example: computing parity

- Input: a binary string, 11010011…111
- Goal: decide whether the number of 1s is odd or even.



In-class exercise

- Draw a complete state-transition table for the parity computing function