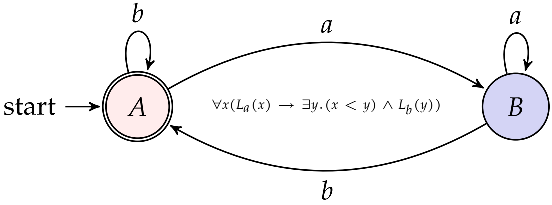
CS 208: Automata Theory and Logic

Lecture 1: An Introduction

Ashutosh Trivedi



Department of Computer Science and Engineering,
Indian Institute of Technology Bombay.

Logistics

- Course Web-page:
  http://www.cse.iitb.ac.in/ trivedi/courses/cs208.html
- Instructors:
  - Ashutosh Trivedi (trivedi@cse)
  - Akshay S. (akshayss@cse)
  - Supratik Chakraborty (supratik@cse)
- Lectures:
  - Tuesday (10am—noon)
  - Friday (10am—noon)
- Tutorials:
  - Wednesday (10am–noon)
- Office hours:
  - Friday (4:00pm–5:00pm)
- Venue
  - Lectures and tutorials: SIC205, 2nd floor, 'C' Block, KReSIT building
  - Office hours: SIA 108, 1st floor, 'A' Block, KReSIT building
- Prerequisite:
  - CS207 (Discrete Structures)
  - Programming experience

Logistics: Contd.

Textbook:

- *E. Hopcroft, R. Motwani and J. D. Ullman*. Introduction to Automata Theory, Languages and Computation. Low priced paperback edition published by Pearson Education.
- *Michael Sipser*. Introduction to the Theory of Computation, PWS Publishing Company.
- *H. R. Lewis and C. H. Papadimitriou*. Elements of the Theory of Computation, Eastern economy edition published by Prentice Hall of India Pvt. Ltd.

Logistics: Contd.

Textbook:

- *E. Hopcroft, R. Motwani and J. D. Ullman*. Introduction to Automata Theory, Languages and Computation. Low priced paperback edition published by Pearson Education.
- *Michael Sipser*. Introduction to the Theory of Computation, PWS Publishing Company.
- *H. R. Lewis and C. H. Papadimitriou*. Elements of the Theory of Computation, Eastern economy edition published by Prentice Hall of India Pvt. Ltd.

Grading:

- End-Semester Exam: 50 %
- Mid-Semester Exam: 30 %
- Surprise Quizzes + Class Participation: 20%

Logistics: Contd.

Textbook:

- *E. Hopcroft, R. Motwani and J. D. Ullman*. Introduction to Automata Theory, Languages and Computation. Low priced paperback edition published by Pearson Education.
- *Michael Sipser*. Introduction to the Theory of Computation, PWS Publishing Company.
- *H. R. Lewis and C. H. Papadimitriou*. Elements of the Theory of Computation, Eastern economy edition published by Prentice Hall of India Pvt. Ltd.

Grading:

- End-Semester Exam: 50 %
- Mid-Semester Exam: 30 %
- Surprise Quizzes + Class Participation: 20%
- Zero tolerance (FR/DAC) for dishonest means like copying solutions from others and cheating.

Introduction to Automata Theory and Logic

Break

Background

Introduction

Dictionary Definition of an Automaton

noun (plural automata)

1. A moving mechanical device made in imitation of a human being.
2. A machine that performs a function according to a predetermined set of coded instructions.

Introduction

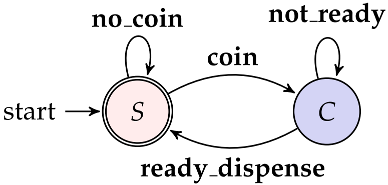
Dictionary Definition of an Automaton

noun (plural automata)

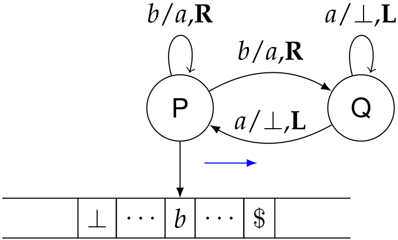
1. A moving mechanical device made in imitation of a human being.
2. A machine that performs a function according to a predetermined set of coded instructions.

Introduction

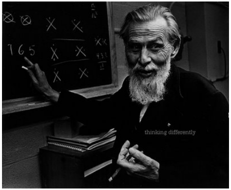
Finite instruction machine with finite memory (Finite State Automata)

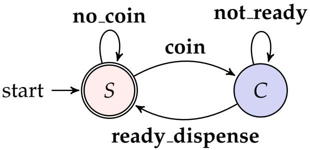


Finite instruction machine with unbounded memory (Turing machine)



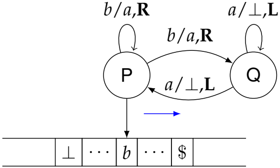
Finite State Automata





- Introduced first by two neuro-psychologist Warren S. McCullough and Walter Pitts in 1943 as a model for human brain!
- Finite automata can naturally model microprocessors and even software programs working on variables with bounded domain
- capture so-called regular sets of sequences that occur in many different fields (logic, algebra, regEx)
- Nice theoretical properties
- Applications in digital circuit/protocol verification, compilers, pattern recognition, etc.

Turing Machine



- Introduced by Alan Turing as a simple model capable of expressing any imaginable computation
- Turing machines are widely accepted as a synonyms for algorithmic computability (Church-Turing thesis)
- Using these conceptual machines Turing showed that first-order logic validity problem [a] is non-computable.
- I.e. there exists some problems for which you can never write a program no matter how hard you try!

[a] (Entscheidungsproblem—one of the most famous problem of 20th century posed by David Hilbert)

What you will learn in this course

- Understand the fundamental notion of computation
- Tools to show that for some problems there are no algorithms (Undecidability)
- Tools to classify practical problems among efficiently solvable versus intractable problems (Complexity classes)
- Ideas behind some of the very useful tools in computer science, like blueefficient pattern matching, syntactic analysis of computer languages, verification of programs (microprocessors, software programs, protocols, etc.)
- The connection between formal logic and automata
- A whole range of formal models of computations (e.g. pushdown automata) between finite state machines and Turing machines with varying expressiveness and efficiency of analysis

Introduction to Automata Theory and Logic

Break

Background

Videos

- Turing Machine by Mike Davey
- LEGO based Turing Machine

Introduction to Automata Theory and Logic

Break

Background

Math-speak

- A set is a collection of objects, e.g.
  - $A = \{a, b, c, d\}$ and $B = \{b, d\}$
  - Empty set $\emptyset = \{\}$ (not the same as $\{\emptyset\}$)
  - $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ and $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
  - $\mathbb{Q}$ is the set of rational number
  - $\mathbb{R}$ is the set of real numbers
- $a \in A$: elements of a set, belongs to, or contains,
- subset of $A \subseteq \mathbb{N}$, proper subset of $A \subset \mathbb{N}$
- Notion of union, intersection, difference, and disjoint
- Power set 2^A of a set A is the set of subsets of A
- Partition of a set

Mathspeak: Contd.

- ordered pair is an ordered pair of elements (a, b), similarly ordered n-tuples, triplets, and so on.
- Cartesian product $A \times B$ of two sets A and B is the set (of tuples) $\{(a, b) \,:\, a \in A \text{ and } b \in B\}$.
- A binary relation R on two sets A and B is a subset of $A \times B$, formally we write $R \subseteq A \times B$. Similarly n-ary relation.
- A function (or mapping) f from set A to B is a binary relation on A and B such that for all $a \in A$ we have that $(a, b) \in f$ and $(a, b') \in f$ implies that $b = b'$.
- We often write $f(a)$ for the unique element b such that $(a, b) \in f$.
- A function $f : A \to B$ is one-to-one if for any two distinct elements $a, a' \in A$ we have that $f(a) \neq f(b)$.
- A function $f : A \to B$ is onto if for every element $b \in B$ there is an $a \in A$ such that $f(a) = b$.
- A function $f : A \to B$ is called bijection if it is both one-to-one and onto B.
- Reflexive, Symmetric, and Transitive relations, and Equivalence relation.

Cardinality of a set

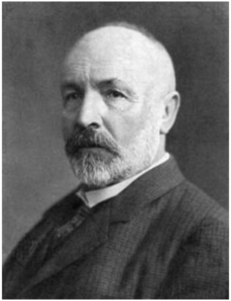
- cardinality $|S|$ of a set S, e.g. $|A| = 4$ and $|\mathbb{N}|$ is an infinite number.
- Two sets have same cardinality if there is a bijection between them.
- A set is countably infinite (or denumerable)if it has same cardinality as $\mathbb{N}$.
- A set is countable if it is either finite or countably infinite.
- A transfinite number is a cardinality of some infinite set.

Let us prove the following.

Theorem

1. *The set of integers is countably infinite.* *(idea: interlacing)*
2. *The union of a finite number of countably infinite sets is countably infinite as well.* *(idea: dove-tailing)*
3. *The union of a countably infinite number of countably infinite sets is countably infinite.*
4. *The set of rational numbers is countably infinite.*
5. *The power set of the set of natural numbers has a greater cardinality than itself.* *(idea: contradiction, diagonalization)*

Cantor's Theorem



Theorem

If a set S is of any infinite cardinality then 2^S has a greater cardinality, i.e. $|2^S| > |S|$. (hint: happy, sad sets).

Corollary ("Most admirable flower of mathematical intellect"—Hilbert.)

There is an infinite series of infinite cardinals. Go figure!

Undecidability

- An alphabet $\Sigma = \{a, b, c\}$ is a finite set of letters,
- A language is a set of strings over some alphabet
- Σ^* is the set of all strings over Σ, e.g. $aabbaa \in \Sigma^*$,
- A language L over Σ is then a subset of Σ^*, e.g.,
  - $L_{\text{even}} = \{w \in \Sigma^* : w \text{ is of even length}\}$
  - $L_{a^n b^n} = \{w \in \Sigma^* : w \text{ is of the form } a^n b^n \text{ for } n \geq 0\}$
- We say that a language L is decidable if there exists a program P_L such that for every member of L program P returns "true", and for every non-member it returns "false".

An intuitive proof for Undecidability

Theorem (Undecidability of Formal Languages)

There are some language for which it is impossible to write a recognizing program, i.e. there are some undecidable formal languages.

An intuitive proof for Undecidability

Theorem (Undecidability of Formal Languages)

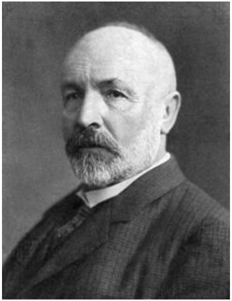
There are some language for which it is impossible to write a recognizing program, i.e. there are some undecidable formal languages.

Proof (Undecidability of formal languages).

- The number of programs are countably infinite, why?
- Consider the set of languages over alphabet $\{0, 1\}$.
- Notice that the set of all strings over $\{0, 1\}$ is countably infinite.
- Hence the set of all languages over $\{0, 1\}$ is the power-set of the set of all strings
- From Cantor's theorem, it must be the case that for some languages there is no recognizing program.

□

Most Admirable, Indeed!



Most admirable, indeed!