Automata theory and its applications

Lecture 1: Historical perspective, course syllabus, basic concepts

Zhilin Wu

State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences

September 26, 2012

Outline

Automata

The origin of the word "automata":

The Greek word "αὐτόματα", which means "self-acting"

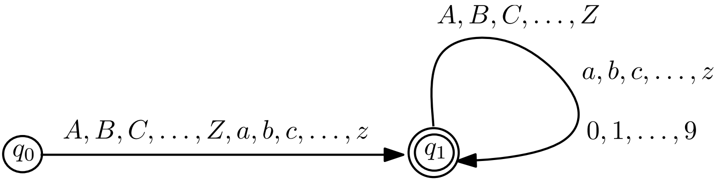
Definition

Abstract models for different aspects of computation

- Sequential computation
  - Finite memory: Finite state automata
  - Finite memory + Stack: Pushdown automata
  - Unrestricted: Turing machines
- Concurrent and reactive systems
  - Nonterminating (ω-words): Büchi automata
  - Nondeterministic (ω-trees): Büchi tree automata
- Rewriting systems
  - Terms: Tree automata over ranked trees
- Semistructured data
  - XML documents: Tree automata over unranked trees

Automata: Example

Finite state automata



Automata for "identifiers" in programming languages

Automata theory

In theoretical computer science, automata theory is

the study of mathematical properties of abstract computing machines.

More specifically

- Expressibility
  Class of languages (computational problems) defined in the model

  What the model
  can and cannot do ?

- Closure properties
  Closed under the different operations, e.g. union and complement.

  The mathematical structure of
  the class of languages defined in the model

- Decidability and complexity
  Are the decision problems (e.g. nonemptiness, inclusion) decidable?
  Can they be solved in PTIME ?

  Are there (efficient) algorithms for
  the statical analysis of the model ?

Outline

Why to bother with automata theory?

- Theoretical foundations of various branches of computer science
  - Origin of computer science

    Turing machine
  - Compiler design

    Lexical analysis (Finite state automata), Syntactical analysis (Pushdown automata), Code selection (Tree automata)
  - Foundations of model checking

    Büchi automata, Rabin tree automata
  - Foundations of Web data (XML document) processing

    Automata over unranked trees
  - ...
- Abstract and fundamental

  Compared to programming languages,
  automata theory is more abstract,
  thus ease the mathematical reasoning,
  but still reflects the essence of computation
- Combinatorial, algorithmic, and challenging

Outline

A pioneer of automata theory

Alan Turing (1912-1954)

- Father of computer science
- English logician
- Propose

  Turing machine
  as
  a mathematical model
  of
  computation
- Codebreaker

  for
  German Enigma machine
  in
  World War II
- Many other pioneering work, e.g. Turing test

Historical perspective of Automata theory

1930s	Turing machines (A. Turing)
1940s -1950s	Finite state automata (W. McCulloch, W. Pitts, S. Kleene, etc.) Chomsky hierarchy (N. Chomsky)
1960s -1970s	Pushdown automata (A.G. Oettinger, M.P. Schutzenberger) Büchi automata over ω-words (J. R. Büchi) Rabin tree automata over ω-trees (M. O. Rabin) Tree automata (J. E. Doner, J. W. Thatcher, J. B. Wright, etc.)
1980s -1990s	ω-automata applied to formal verification (M. Vardi, P. Wolper, O. Kupferman, etc.)
2000s-2010s	Automata over unranked trees applied to XML (A. Bruggemann-Klein, M. Murata, D. Wood, F. Neven, etc.) Visibly pushdown automata (R. Alur, P. Madhusudan)

Remark: Only include automata models explicitly referred to in this course

Outline

Goal, organization, ...

Main goal

An extensive introduction to automata theory,
with an emphasis on the developments in the last thirty years.

Organization

Organized according to the different types of structures,
Finite words, Infinite words, Finite ranked and unranked trees, Infinite trees

Features

- Extensiveness of the topics
  In particular, the recent developments are covered
- Emphasis on algorithmic aspects
- Emphasis on the applications

Not included

- Automata models for timed and hybrid systems
  Timed automata, Hybrid automata
- Automata over infinite alphabets
  Register automata, data automata, ...

Syllabus (tentative and possibly adjusted later)

- Automata over finite words
  - Chomsky hierarchy (A brief recall of the classical automata theory)
    - Turing machines, Linearly-bounded automata,
    - Pushdown automata, Finite state automata
  - Finite state automata
    - Nondeterministic versus deterministic, Expressive equivalence with MSO,
    - Myhill-Nerode theorem, Closure properties,
    - Decision problems (Nonemptiness, language Inclusion)
  - Visibly pushdown automata
    - Nondeterministic versus deterministic,
    - Closure properties,
    - Decision problems
- Automata over infinite words
  - Nondeterministic Büchi, Muller, Rabin, Strett, Parity automata and their expressive equivalence
  - Expressive equivalence with MSO and WMSO
  - Determinization and complementation of Büchi automata
  - Decision problems of Büchi automata

Syllabus: continued

- Automata over finite trees
  - Ranked trees

    Bottom-up versus top-down tree automata, Expressive equivalence with MSO, Determinization, Decision problems, Tree-walking automata
  - Unranked trees

    The model, Expressive equivalence with MSO, Determinization, Decision problems
- Automata over infinite trees
  - Büchi tree automata, Rabin tree automata, Parity tree automata, and the comparison of their expressibility
  - Complementation of Rabin tree automata
  - The decision problems

Syllabus: continued

- Applications
  - Model checking
    - Linear temporal logic and Büchi automata
    - Alternating tree automata, Modal μ-calculus
  - XML document processing
    - XPath, DTD, and their relationship with automata over unranked trees
    - Visibly pushdown automata applied to the streaming of XML documents

Grading criteria

- Homework (not so many, do not worry): 20%
- Reading project: 20%
  Read a paper from a given list and make a presentation
- Reading report: 60%
  Choose a topic from a given list and make a survey

Remark: The paper list and the topic list will be published later.

References

Course website: http://lcs.ios.ac.cn/∼wuzl/teaching.html

- Textbooks
  1. D. Kozen, *Automata and computability*, Springer, 1997.
  2. B. Khoussainov, A. Nerode, *Automata theory and its applications*, Springer, 2001.
  3. E. Gradel, W. Thomas, T. Wilke, *Automata, logics and infinite games*, Spirnger, 2002.
  4. H. Comon and M. Dauchet and R. Gilleron and C. Löding and F. Jacquemard and D. Lugiez and S. Tison and M. Tommasi, *Tree Automata Techniques and Applications*, http://tata.gforge.inria.fr/
- Survey articles
  1. W. Thomas, *Languages, automata and logics*, Handbook of formal languages, vol. 3, Pages 389 - 455.
  2. T. Schwentick, *Automata for XML–A survey*, Journal of Computer and System Sciences, vol. 73(3), 2007, Pages 289-315.
- Possibly some other papers

Remark: You are not supposed to read all of them.

Outline

Finite and infinite words

Fix a finite alphabet Σ.

- Finite words

  A sequence of letters from Σ, i.e. a mapping $w : [n] \to \Sigma$.
  Example: abab

- ω-words

  An ω-sequence of letters from Σ, i.e. a mapping $w : \mathbb{N} \to \Sigma$.
  Example: $(ab)^\omega$

Finite and infinite trees

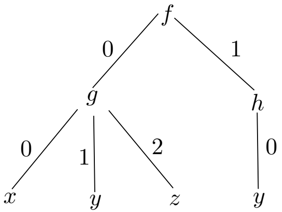
- Finite ranked trees

  Ranked alphabet Σ*: Rank function rank():* $\Sigma \to \mathbb{N}$.
  Tree domain: A nonempty subset D *of* $\mathbb{N}$ *such that*

  - *if* $xi \in D$, *then* $x \in D$,
  - *if* $xi \in D$, *then* $xj \in D$ *for any* $j \leq i$.

  Ranked trees: A Σ*-tree is a mapping* $t : D \to \Sigma$ *such that*

$$\forall x \in D, rank(t(x)) = |\max\{i \mid xi \in D\}|.$$

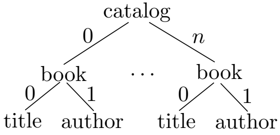


Finite and infinite trees: continued

- Finite unranked trees

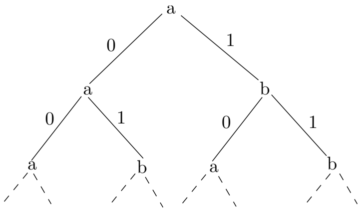
  Alphabet Σ *is unranked*

  Unranked trees: A mapping $t : D \to \Sigma$ *(no rank constraints).*



- ω-trees (binary ω-trees)

  A mapping $t : \{0,1\}^* \to \Sigma$.



$t(\varepsilon) = a$

$t(x0) = a$

$t(x1) = b$

Formal languages and closure properties

- Formal languages

  A set of finite words, finite trees, etc.

- Language-theoretical operations
  - Union: $L_1 \cup L_2$,
  - Intersection: $L_1 \cap L_2$,
  - Complementation: $\Sigma^* \setminus L$, $\Sigma^\omega \setminus L$, ...
  - Homomorphism: A mapping $h : \Sigma \to \Pi \cup \{\varepsilon\}$.
- Closure properties
  - Union:
    For every pair of automata $\mathcal{A}_1$ and $\mathcal{A}_2$ in a given model,
    is $\mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$ also accepted by an automaton in the model ?
  - and so on

Example:

Finite state automata are closed under all Boolean operations (union, intersection and complementation).

Expressibility and decision problems

- Expressibility: Which languages can be defined in the model ?
  Example

  The language $\{a^n b^n \mid n \in \mathbb{N}\}$
  cannot be defined by finite state automata.

- Decision problems
  - Nonemptiness

    Given an automaton $\mathcal{A}$*, does* $\mathcal{L}(\mathcal{A}) \neq \emptyset$ *?*

  - Universality

    Given an automaton $\mathcal{A}$*, does* $\mathcal{L}(\mathcal{A}) = \Sigma^*$ *(for finite words) ?*
    Similarly for other classes of structures

  - Language inclusion

    Given two automata $\mathcal{A}_1, \mathcal{A}_2$*, does* $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ *?*