ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Automata and Formal Languages

Automata and Formal Languages

- What is a Finite Automaton?
- Why study Automata?
- What the course is about?

What is a Finite Automaton?

- Remembers only a finite amount of information.
- Information represented by its *state*.
- State changes in response to *inputs*.
- Rules that tell how the state changes in response to inputs are called *transitions*.

What is a Finite Automaton?

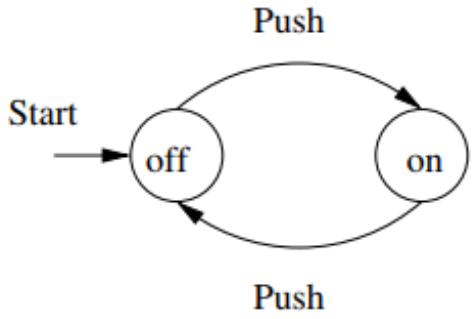

Figure 1.1: A finite automaton modeling an on/off switch

What is a Finite Automaton?

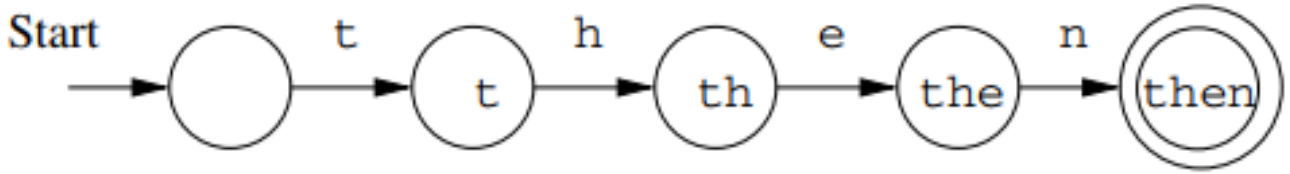

Figure 1.2: A finite automaton modeling recognition of then

Why study Automata?

Why study Automata?

- Regular expressions are used in many systems.
  - E.g., UNIX [A-Z][a-z]*[ ][A-Z][A-Z].
    - This expression represents patterns in text that could be a city and state, e.g. **Ithaca NY**
  - E.g., DTD's describe XML tags with a RE format like person (name, addr, child*).

Why study Automata?

- Finite automata model protocols, electronic circuits.

Why study Automata?

- Context-free grammars are used to describe the syntax of essentially every programming language.
  - Not to forget their important role in describing natural languages.

Why study Automata?

- When developing solutions to real problems, we often confront the limitations of what software can do.
  - *Undecidable* things – no program whatever can do it.
  - *Intractable* things – there are programs, but no fast programs.

Course Outline

- Finite Automata
- Regular Expressions
- Context-Free Grammars
- Pushdown Automata
- Turing Machines

The Central Concepts of Automata Theory

- Alphabets
  - An alphabet is a finite nonempty set of symbols.
  - Conventionally we use the symbol $\sum$ for an alphabet
  - Common alphabets include:
    - $\sum \equiv \{0,1\}$, the binary alphabet
    - $\sum \equiv \{a, b, \ldots, z\}$, the set of all lower-case letters
    - The set of all ASCII characters, or the set of all printable ASCII characters

The Central Concepts of Automata Theory

- Strings
  - A **string** (or sometimes word) is a finite sequence of symbols chosen from some alphabet.
    - `01101` is a string from the binary alphabet $\sum \equiv \{0,1\}$
    - `111` is another string chosen from this alphabet
  - The Empty String: the string with zero occurrences of symbols.
    - denoted **ε**
    - is a string that may be chosen from any alphabet whatsoever
  - Length of a String: the number of positions for symbols in the string
    - string `01101` has length `5`
    - the standard notation for the length of a string ω is **|ω|**
    - `|011| = 3` và `|ε| = 0`

The Central Concepts of Automata Theory

- Strings
  - Powers of an Alphabet
    - If Σ is an alphabet, we can express the set of all strings of a certain length from that alphabet by using an exponential notation.
    - Σ^k to be the set of strings of length k, each of whose symbols is in Σ
    - $\Sigma^0 = \{\varepsilon\}$, regardless of what alphabet Σ is
      - that is ε is the only string whose length is 0
      - $\Sigma \equiv \{0,1\}$: $\Sigma^1 = \{0, 1\}$, $\Sigma^2 = \{00, 01, 10, 11\}$, $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$, …….
    - Σ^* : the set of all strings over an alphabet Σ
      - $\{0,1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots..\}$
      - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots\cdots\cdots$
    - Σ^+: set of nonempty strings from alphabet Σ
      - two appropriate equivalences
        - $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots\cdots\cdots$
        - $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$

The Central Concepts of Automata Theory

- Strings
  - Concatenation of Strings
    - Let *x* and y be strings.
    - Then xy denotes the concatenation of x and y
      - the string formed by making a copy of x and following it by a copy of y
      - let x = 01101 and y = 110, then xy = 01101110 and yx = 11001101

The Central Concepts of Automata Theory

- Languages
  - A set of strings all of which are chosen from some Σ^*, where Σ is a particular alphabet is called a language.
    - If Σ is an alphabet and $L \subseteq \Sigma^*$ then L is a language over Σ
    - Notice that a language over Σ need not include strings with all the symbols of Σ
  - There are also many other languages that appear when we study automata
    - The language of all strings consisting of n 0's followed by n 1's for some $n \geq 0$: {ε, 01, 0011, 000111, …..}
    - The set of strings of 0's and 1's with an equal number of each: {ε, 01, 10, 0011, 0101, 1001, …..}
    - The set of binary numbers whose value is a prime: {10, 11, 101, 111, 1011, …..}
    - Σ^* is a language for any alphabet Σ
    - $\varnothing$, the empty language, is a language over any alphabet
    - $\{\varepsilon\}$, the language consisting of only the empty string, is also a language over any alphabet.
      - Notice that $\varnothing \neq \{\varepsilon\}$; the former has no strings and the latter has one string.

Problems

- If $\sum$ is an alphabet, and L is a language over $\sum$ then the problem L is:
  - Given a string w in $\sum^*$, decide whether or not w is in L.

Set-Formers as a Way to Define Languages

- It is common to describe a language using a "set-former"

  {w | something about w}

This expression is read "the set of words w such that (whatever is said about w to the right of the vertical bar)"

  - {w | w consists of an equal number of 0's and 1's }
  - {w | w is a binary integer that is prime }
  - {w | w is a syntactically correct C program }

- It is also common to replace w by some expression with parameters and describe the strings in the language by stating conditions on the parameters
  - $\{0^n1^n \mid n \geq 1\}$
  - $\{0^i1^j \mid 0 \leq i \leq j\}$