Automata Theory

• Automata theory is the study of abstract computing devices.
• A. M. Turing studied an abstract machine that had all the capabilities of today’s computers.
• Turing’s goal was to describe the boundary between what a computing machine could do and what it could not do.
Finite Automata

• Studied 40’s and 50’s
• Originally proposed to model brain function
• Turned out to be useful for other purposes
What is finite automata for?

• Designing and checking the behavior of digital circuits
• Lexical analyzer of a typical compiler, that is, compiler component that breaks the input text into logical units
• Scanning large bodies of text
• Verifying systems of all types that have a finite number of distinct states
• Etc.
Simplest nontrivial finite automaton remembering whether it is in the “on” state or the “off” state
Automaton recognizing “then”
Why study Automata theory?

• What can a computer do at all? This study is called “decidability” and the problems that can be solved by computer are called “decidable”. (chapter 9)

• What can computer do efficiently? This study is called “intractability”, and the problems that can be solved by a computer using no more time than some slowly growing function (polynomial) of the size of the input are called “tractable”.

Review of Basics

Sets – Definition
A set is an unordered collection of objects

\[ S = \{a, b, c\} \]

a, b, c are elements or members of the set S
Sets – Definition
A set is an unordered collection of objects

\[ S = \{a, b, c\} \]

\(a, b, c\) are elements or members of the set \(S\)

Elements in a set need have no relation to each other

- \(S_1 = \{1, 2, 3\}\)
- \(S_2 = \{\text{red, farmhouse, } \pi, -32\}\)
Sets – Definition

Sets can contain other sets as elements

• $S_1 = \{3, \{3, 4\}, \{4, \{5, 6\}\}\}$
• $S_2 = \{\{1, 2\}, \{\{4\}\}\}$

Sets do not contain duplicates
NotASet = \{4, 2, 4, 5\}
Sets – Cardinality

Cardinality of a set is the number of elements in the set

\[ |\{a, b, c\}| = 3 \]
\[ |\{\{a, b\}, c\}| = ? \]
Sets – Cardinality
Cardinality of a set is the number of elements in the set

- \(|\{a, b, c\}| = 3\)
- \(|\{\{a, b\}, c\}| = 2\) (\({a, b}\) and \(c\))
Sets – Empty, Singleton
Empty Set: {} or Ø , |{}| = |Ø| = 0

Singleton set – set with one element

•{1}
•{4}
•{} ?
•{{}} ?
•{{3, 1, 2}} ?
Sets – Empty, Singleton

Empty Set: {} or $\emptyset$ , $|\{}| = |\emptyset| = 0$

Singleton set – set with one element

• {1}
• {4}
• {} NO! Set contains 0 elements
• {}{}
• {{3, 1, 2}}
Sets – Membership

Set membership: \( x \in S \)

\( 3 \in \{1, 3, 5\} \)

\( a \notin \{b, c, d\} \)

\( 3 \in \{1, \{2, 3\}\} \) ?

\( \emptyset \in \{1, 2, 3\} \) ?

\( \emptyset \in \{1, \emptyset, 4\} \) ?
Sets – Membership

Set membership: $x \in S$

$3 \in \{1, 3, 5\}$

$a \notin \{b, c, d\}$

$3 \notin \{1, \{2, 3\}\}$

$\emptyset \notin \{1, 2, 3\}$

$\emptyset \in \{1, \emptyset, 4\}$?
Sets – Describing

$S = \{ x : x \text{ has a certain property} \}$

$S = \{ x | x \text{ has a certain property} \}$

- $S = \{ x : x \in \mathbb{N} \land x < 10 \}$
  - $\mathbb{N}$ is the set of natural numbers $\{0, 1, 2, \ldots \}$

- $S = \{ x : x \text{ is prime } \}$
- $A \cap B = \{ x : x \in A \land x \in B \}$
- $A \cup B = \{ x : x \in A \lor x \in B \}$
- $A - B = \{ x : x \in A \land x \notin B \}$
• \( S = \{0^n1^n \mid n \geq 1\} \)
• \( S = \{01, 0011, 000111, \ldots\} \)
• \( T = \{0^i1^j \mid 0 \leq i \leq j\} \)
• \( T = \{1, 11, \ldots, 01, 011, \ldots, 0011, 00111, \ldots\} \)
Sets – ∩, ∪

More Union & Intersection
- A and B are **disjoint** if \( A \cap B = \emptyset \)
- S is a collection of sets (set of sets)
  \[ \bigcup S = \{x : x \in A \text{ for some } A \in S\} \]
  - \( \bigcup \{\{1, 2\}, \{2, 3\}\} = \{1, 2, 3\} \)
  - \( \bigcap S = \{x : x \in A \text{ for all } A \in S\} \)
  - \( \bigcap \{\{1, 2\}, \{2, 3\}\} = \{2\} \)
Sets – Subset

• Subsets & Supersets

  • A is a subset of B, \( A \subseteq B \) if:
    • \( \forall x, x \in A \Rightarrow x \in B \)
    • \( \forall (x \in A), x \in B \)
  • A is a proper subset of B, \( A \subset B \) if:
    • \( A \subseteq B \land (\exists x, x \in B \land x \notin A) \)
  • \( \{\} \) is a subset of any set (including itself)
  • \( \{\} \) is the only set that does not have a proper subset
Sets – Power Set

• Power set: Set of all subsets
• \( 2^S = \{x : x \subseteq S\} \)
  – \( 2^{\{a,b\}} = \{\emptyset, \{a\}, \{b\}, \{a, b\}\} \)
  – \( 2^\emptyset = \{\emptyset\} \)
• \( |2^S| = ? \) (cardinality?)
• \( |2^S| = 2^{|S|} \)
Cartesian Product

\[ A \times B = \{(x, y) : x \in A \land y \in B\} \]

- \{1, 2\} \times \{3, 4\} = \{(1, 3), (1, 4), (2, 3), (2, 4)\}
- \{1, 2\} \times \{1, 2\} = \{(1, 1), (1, 2), (2, 1), (2, 2)\}
- \{a\} \times \{b\} = \{\{(a, b)\}, \{\}\}\n- \{a\} \times \{b\} = \{\{\{a\}, \{b\}\}, \{\}, \{\{a\}\}, \{\{b\}\}, \{\}, \{\}\}\}
Cartesian Product

Which of the following is true:

• $\forall (A,B) \quad A \times B = B \times A$
  • If and only if $A = B$
• $\forall (A,B) \quad A \times B \neq B \times A$
  • If and only if $A \neq B$
Relations

A relation $R$ is a set of ordered pairs.

For example the relation ‘$<$‘ over the Natural Numbers is the set:

$$\{(0,1), (0,2), (0,3), \ldots$$

$$ (1,2), (1,3), (1,4), \ldots$$

$$ (2,3), (2,4), (2,5), \ldots$$

$$ \ldots \}$$

Often, relations are over the same set that is, a subset of $A \times A$ for some set $A$.

Not all relations are over the same set, however.

Relation describing prices of computer components:

$$\{(\text{Hard dive}, \$55), (\text{WAP}, \$49), (256\text{M DDR}, \$44), \ldots\}$$
Functions

• A **function** is a special kind of relation (all functions are relations, but not all relations are functions)
• A relation $R \subseteq A \times B$ is a function if:
  – For each $a \in A$, there is exactly one ordered pair in $R$ with the first component $a$
• A function $f$ that is a subset of $A \times B$ is written: $f : A \rightarrow B$
  – $(a, b) \in f$ is written $f(a) = b$
  – $A$ is the domain of the function
  – if $A' \subseteq A$, $f(A') = \{b : a \in A' \land f(a) = b\}$ is the **image** of $A'$
  – The **range** of a function is the image of its domain
Functions

A function $f : A \rightarrow B$ is:

- **one-to-one** if no two elements in $A$ match to the same element in $B$
- **onto** Each element in $B$ is mapped to by at least one element in $A$
- **a bijection** if it is both one-to-one and onto

The **inverse** of a binary relation $R \subseteq A \times B$ is denoted $R^{-1}$, and defined to be $\{(b, a) : (a, b) \in R\}$

- A function only has an inverse if it is a bijection
Structural representations

• Grammars (will be discussed in ch5)
  – $E \Rightarrow E + E$  an expression can be formed by taking any two expressions and connecting them by a plus sign

• Regular Expressions
  – $([A-Z][a-z]*[ ])*[ ][A-Z][A-Z]$  Now matches with ‘Palo Alto CA’
Central Concepts of Automata Theory

- Alphabet: finite, non empty set of symbols
- Strings: list of symbols from the alphabet
- Language: a set of strings from the same alphabet.
Alphabets

Conventionally denoted by ‘Σ’

Examples of common alphabets:

1. $\Sigma = \{0,1\}$, the binary alphabet
2. $\Sigma = \{a, b, c, \ldots, z\}$, the set of all lower-case letters
3. The set of ASCII characters…
Strings

• A string (or word) is a finite sequence of symbols chosen from some alphabet.
  – 00110101 from binary alphabet $\Sigma=\{0,1\}$
• The empty string is the string with zero occurrences of symbols. Denoted by ‘$\varepsilon$’.
• The number of positions for symbols (ie, number of symbols) in the strings is called the length of the string.
  – ‘00110101’ has length 8.
• The standard notation for the length of a string $\omega$ is $|\omega|$.
  – $|00110101| = 8$
  – $|\varepsilon| = 0$
Powers of an alphabet

If $\Sigma$ is an alphabet, we define $\Sigma^k$ to be the set of strings of length $k$, each of those symbols in $\Sigma$.

Examples:

• $\Sigma^0 = \{\varepsilon\}$ for all alphabets

• If $\Sigma = \{0,1\}$
  - $\Sigma^1 = \{0,1\}$
  - $\Sigma^2 = \{00,01,10,11\}$
  - $\Sigma^3 = \{000,001,010,011,100,101,110,111\}$
• $\Sigma^*$ denotes set of all strings over an alphabet $\Sigma$.

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$

• $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$

  Same as previous one but without empty string.

• $\Sigma^* = \Sigma^0 \cup \Sigma^+$
Type convention for symbols and strings

• We shall use lower case letters at the beginning of the alphabet to denote symbols
  – a, b, c, a₁, a₂, c₃,…
• lower case letters at the end of the alphabet to denote strings
  – x, y, z
Concatanation of strings

• Let $x$ and $y$ be strings. Then $xy$ denotes the concatenation of $x$ and $y$.

Example:
    let $x = 01011$ and $y = 011$.
    Then $xy = 01011011$
Languages

A set of strings all of which are chosen from some $\Sigma^*$, where $\Sigma$ is a particular alphabet, is called language.

• If $\Sigma$ is an alphabet, and $L \subseteq \Sigma^*$, then $L$ is language over $\Sigma$. 
Language examples

• Turkish
• C
• The language of all strings consisting of n 0’s followed by n 1’s, for some n ≥ 0:
  \{ε, 01, 0011, 000111, 00001111, ……\}
• The set of binary numbers whose value is a prime
  \{10,11,101,111,1011,…\}
• \(\Sigma^*\) (set of all strings)
• \(\emptyset\), 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.
Problems

In Automata theory, a *problem* is the question of deciding whether a given string is a member of some particular language.

If $\Sigma$ is an alphabet, and $L (\subseteq \Sigma^*)$ is a language over $\Sigma$, then the problem is:

- Given a string $\omega$ in $\Sigma^*$, decide whether or not $\omega$ is in $L$. 
Prime numbers example

The problem of testing primality can be reduced to a problem of testing whether the given number is a member of $L_p$ (language of prime numbers) or not.
Definition of problems as languages

• We are interested in proving lower bounds on the complexity of certain problems.
• Especially important are techniques for proving that certain problems cannot be solved in an amount of time that is less than exponential in the size of the input.