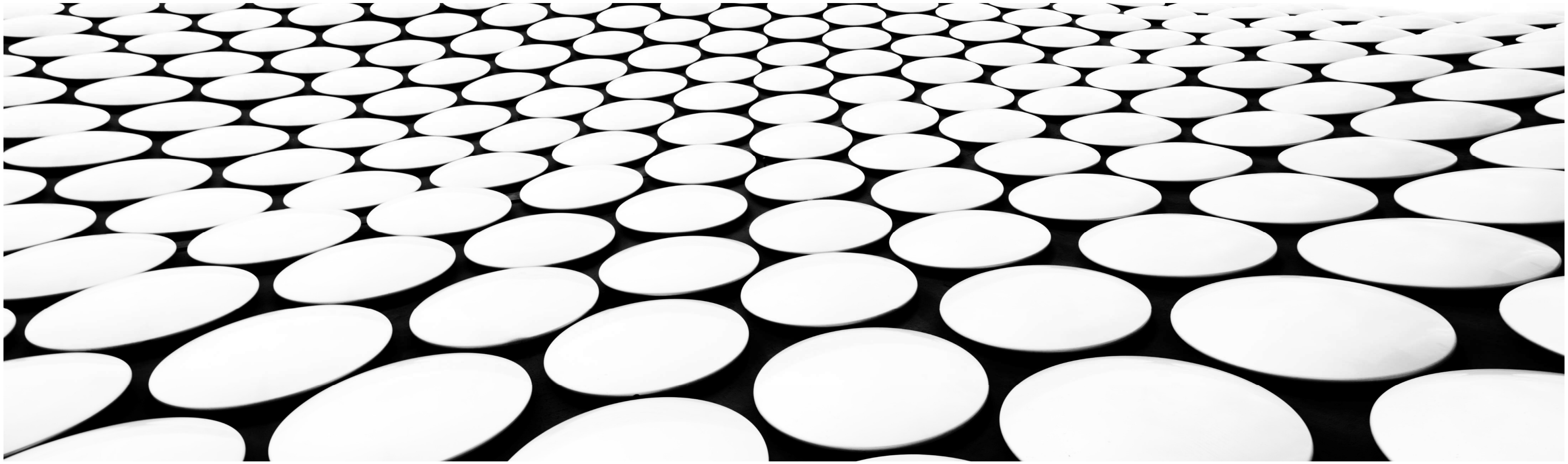
FINITE AUTOMATA

IF 2124 TEORI BAHASA FORMAL OTOMATA

Judhi S.



Motivation

- Automata = abstract computing devices

- Turing studied Turing Machines (= computers) before there were any real computers

- We will also look at simpler devices than Turing machines (Finite State Automata, Pushdown Automata, . . .), and specification means, such as grammars and regular expressions.

- NP-hardness = what cannot be efficiently computed

Finite Automata

Finite Automata are used as a model for

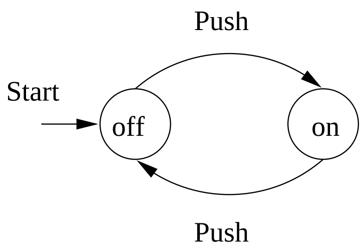
- Software for designing digital cicuits

- Lexical analyzer of a compiler

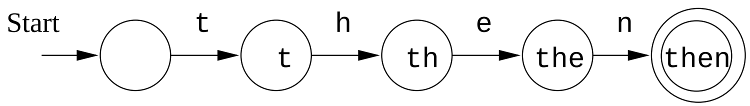
- Searching for keywords in a file or on the web.

- Software for verifying finite state systems, such as communication protocols.

- Example: Finite Automaton modelling an on/off switch



- Example: Finite Automaton recognizing the string `then`



Structural Representations

These are alternative ways of specifying a machine

Grammars: A rule like $E \Rightarrow E + E$ specifies an arithmetic expression

- $Lineup \Rightarrow Person.Lineup$

says that a lineup is a person in front of a lineup.

Regular Expressions: Denote structure of data, e.g.

`'[A-Z][a-z]*[][A-Z][A-Z]'`

matches `Ithaca NY`

does not match `Palo Alto CA`

Question: What expression would match

`Palo Alto CA`

Central Concepts

Alphabet: Finite, nonempty set of symbols

Example: $\Sigma = \{0, 1\}$ binary alphabet

Example: $\Sigma = \{a, b, c, \ldots, z\}$ the set of all lower case letters

Example: The set of all ASCII characters

Strings: Finite sequence of symbols from an alphabet Σ, e.g. 0011001

Empty String: The string with zero occurrences of symbols from Σ

- The empty string is denoted ϵ

Length of String: Number of positions for symbols in the string.

$|w|$ denotes the length of string w

$|0110| = 4, |\epsilon| = 0$

Powers of an Alphabet: Σ^k = the set of strings of length k with symbols from Σ

Example: $\Sigma = \{0, 1\}$

$\Sigma^1 = \{0, 1\}$

$\Sigma^2 = \{00, 01, 10, 11\}$

$\Sigma^0 = \{\epsilon\}$

Question: How many strings are there in Σ^3

The set of all strings over Σ is denoted Σ^*

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$

Also:

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots$

$\Sigma^* = \Sigma^+ \cup \{\epsilon\}$

Concatenation: If x and y are strings, then xy is the string obtained by placing a copy of y immediately after a copy of x

$x = a_1 a_2 \ldots a_i, y = b_1 b_2 \ldots b_j$

$xy = a_1 a_2 \ldots a_i b_1 b_2 \ldots b_j$

Example: $x = 01101, y = 110, xy = 01101110$

Note: For any string x

$$x\epsilon = \epsilon x = x$$

Languages:

If Σ is an alphabet, and $L \subseteq \Sigma^*$
then L is a language

Examples of languages:

- The set of legal English words

- The set of legal C programs

- The set of strings consisting of n 0's followed by n 1's

$$\{\epsilon, 01, 0011, 000111, \ldots\}$$

- The set of strings with equal number of 0's and 1's

$$\{\epsilon, 01, 10, 0011, 0101, 1001, \ldots\}$$

- $L_P =$ the set of binary numbers whose value is prime

$$\{10, 11, 101, 111, 1011, \ldots\}$$

- The empty language $\emptyset$

- The language $\{\epsilon\}$ consisting of the empty string

Note: $\emptyset \neq \{\epsilon\}$

Note2: The underlying alphabet Σ is always finite

Problem: Is a given string w a member of a language L?

Example: Is a binary number prime = is it a meber in L_P

Is $11101 \in L_P$? What computational resources are needed to answer the question.

Usually we think of problems not as a yes/no decision, but as something that transforms an input into an output.

Example: Parse a C-program = check if the program is correct, and if it is, produce a parse tree.

Let L_X be the set of all valid programs in prog lang X. If we can show that determining membership in L_X is hard, then parsing programs written in X cannot be easier.

Question: Why?

Finite Automata Informally

Protocol for e-commerce using e-money

Allowed events:

1. The customer can *pay* the store (=send the money-file to the store)

2. The customer can *cancel* the money (like putting a stop on a check)

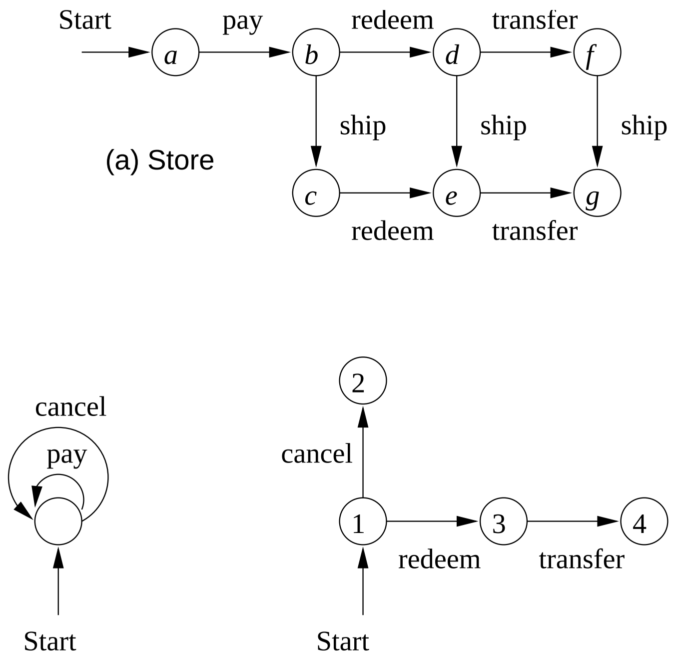
3. The store can *ship* the goods to the customer

4. The store can *redeem* the money (=cash the check)

5. The bank can *transfer* the money to the store

e-commerce

The protocol for each participant:

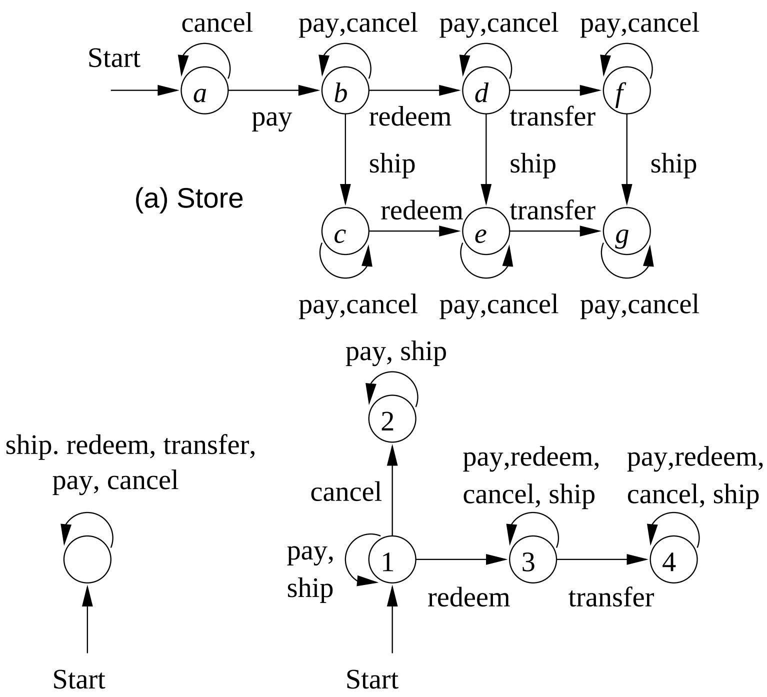


(a) Store

(b) Customer

(c) Bank

Completed protocols:



(b) Customer (c) Bank

product 2 automata seperti irisan

The entire system as an Automaton: