Information Technology
Inside and Outside
- David Cyganski & John A. Orr

II. Fundamentals of Binary Representation

3. Representing Information in Bits

Hoon-Jae Lee
hjlee@dongseo.ac.kr
http://cg.dongseo.ac.kr/~hjlee

3. Representing Information in Bits

- The fact that the same information may be represented in many different ways, by a variety of physical or logical elements;
- the concept of a "code" for information, with examples such as the Roman alphabet and the Chinese character set;
- the binary number system and the means by which all information can be represented by codes containing only zeroes and ones;
- specific examples of the representation of numeric and text data with binary digits (bits);
- the properties of signals that vary with time, such as sound, and of image signals that require two-dimensional representation; and
- the means by which errors in stored or transmitted information may be detected and corrected.

3.1 Introduction
3.2 Information and Its Representation

- A technique for representing information
  - the technique must allow us to ***uniquely*** represent information and to recreate it in its original form;
  - the technique must be ***standardized*** so that it can be used for many different applications: numerical data, text, audio, still and moving images, and more; and
  - the technique must be ***compatible*** with inexpensive and reliable technology for handling the information.

3.3 The Search for an Appropriate Code

- ***Codes*** with a finite number of basic elements, called an ***alphabet***
  - Codes have a limited number of different ***symbols***

3.3.1 A Look at Written Alphabets

- The alphabet used for the written **English** language is commonly thought to contain about **96 elements**
  - 26 lower case characters, 26 upper case characters
  - 10 numbers, and
  - 32 special characters, such as a space or a dollar sign
  - How ``good'' is this code for representing information?
- The Mandarin profile of the **Chinese**: The system was developed over 4,000 years ago. It uses a set of logo graphs(characters) of several types: pictographs, ideographs, compound ideographs, loan characters, and phonetic compounds. The latter forms over 90 percent of the total set of as many as 40,000 characters.
  - Certainly this is a``powerful'' code; one complex character can convey an entire concept to the skilled reader.

3.3 The Search for an Appropriate Code(2)

3.3.1 A Look at Written Alphabets (2)

Figure 3.1: Chinese characters and their English translation.

- ***Fewer characters*** are needed to communicate a set of ideas than if we were to use the letters of written English.
- Written form of Chinese is thought of as one of the most ``difficult'' written languages to use.
- English letters are fewer in number, and therefore each one ***conveys less information*** than a Chinese character.
- But this also makes English letters ***simpler to distinguish from one another***, and thus, less likely to cause misinterpretation.

3.3 The Search for an Appropriate Code(3)

3.3.2 The Need for a Robust Scheme

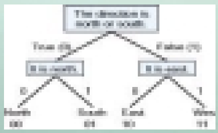
- **In the design of our code**, we must consider the question of which is more important to us: that **each symbol of the code convey a lot of information**, or that **we be able to readily distinguish the symbols from each other**.
  - The answer lies in our requirement for **an information code that is compatible with inexpensive and reliable equipment for handling it.**
- Reliable manipulation of information depends upon ***tolerance to errors.***
- Information code must represent information in a way that is ***robust***, or tolerant to errors.
- The fewer symbols the code has, the easier it is to distinguish the symbols from each other, and the more robust the code will be.
- A code with just two symbols, called a ***binary code***, might at first seem to be almost as useless.

3.4 Bits as Building Blocks of Information

- A common way of representing the two different symbols in a binary code is by using the first two integers, ``0" and ``1."
  - ``T" and ``F," / ``true" and ``false”

3.4.1 The Representational Power of Bits

- How much information can we convey with one binary symbol? We can convey the answer to a single true/false question.
- A two-bit word can be arranged in any of four patterns: 00, 01, 10,or 11. Thus, by using two bits (or equivalently, by answering two true/false questions) we can represent any one of four different things.



3.4 Bits as Building Blocks of Information(2)

3.4.1 The Representational Power of Bits(2)

- A pattern of two bits to represent anyone of the four directions:
  - 00: North
  - 01: South
  - 10: East
  - 11: West
- Adding a third bit increases the repersentational power of our bit string to one of eight patterns: 000, 001, 010, 011, 100, 101, 110, and 111. We could use one of these three-bit words to represent any of Santa's eight reindeer (not including Rudolph).
- In general, if we have ***n*** bits in the codeword, then there are 2^n different codewords, which can represent one of 2^n different messages.
- For example, 8 bits can be assembled in $2^8 = 256$ different patterns, and thus can represent any one of 256 different messages.

3.4 Bits as Building Blocks of Information(3)

3.4.2 Bits in the Physical World

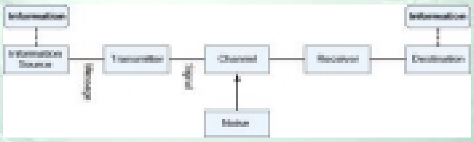
- Some familiar technologies for handling information reveals that a binary code is well suited to practical equipment.
  - Storage Equipment
  - Transmission Equipment:
  - Processing Equipment

- Storage Equipment
  - A **magnetic disk**, which has many small areas, called ***domains***, each of which can store a single bit of information. Each domain can be magnetized in one of two directions--**``up'' or ``down''**--corresponding to whether the bit to be stored is **a 0 or a 1**.
  - **CD** consists of many tiny ***domains***,each of which stores one bit of information. In each domain, there is either a smooth surface, which will **reflect** a laser, **or** a pit,which will **not**.

3.4 Bits as Building Blocks of Information(4)

3.4.2 Bits in the Physical World(2)

- Transmission Equipment:



  - **An information transmission system**, at its simplest, consists of a ***transmitter***, a ***channel*** over which the information travels, and a ***receiver*** at the destination.
  - Common examples of transmission channels are **wires, electrical cables, optical fibers, and even the air** in the case of information broadcast in the form of radio waves.
  - The goal of this system is for information to be **transferred, without loss or modification**, from the transmitter to the receiver

3.4 Bits as Building Blocks of Information(5)

3.4.2 Bits in the Physical World(3)

- Transmission Equipment(2):
  - Unfortunately, the channel of an information transmission system typically is subject to ***noise***, or unwanted and unpredictable interference from external sources.
  - This may come from other information systems, or from natural sources such as lightning or radiation.
  - In a binary system, the receiver must simply interpret each received symbol as a ``0'' or a ``1.'' That is, it does not matter whether a binary message has been distorted, as long as the receiver can still distinguish the message ``0'' from the message ``1.'' In this sense, the system is ***immune*** to moderate amounts of noise or disturbance.

3.4 Bits as Building Blocks of Information(6)

3.4.2 Bits in the Physical World(4)

- Processing Equipment
  - ***Processing*** may be intended to **modify the actual information** content, just as an editor modifies the content of a document. Or, the processing may be intended to **``repackage'' the information** in a form more suitable to transmission or storage.
  - Computer circuits can be broken down into simpler and simpler sub circuits until we reach the fundamental building block from which the entire computer is made. This building block is the electronic switch. The electronic switch is a circuit designed always to be in one of two states: *ON* or *OFF*. These states are similar to the two positions of a physical switch.
  - ***High-level* complexity** in computer system is made possible **by the simplicity and reliability of its *low-level* operations.**

3.4 Bits as Building Blocks of Information(7)

3.4. 3 From Numbers to Bits

- **Integer Formats**
  - an ***n***-bit word can represent 2^n different things.
  - $2^{16} = 65,536$
  - We could represent each of the integers from 1 to 65,536 (or 0 to 65,535) by using a different 16-bit word.
  - 16 bits to represent integers between -32,768 and 32,767 (including zero).
  - Decimal to Binary conversion applet:

  - Binary to Decimal conversion applet:

3.4 Bits as Building Blocks of Information(8)

3.4. 3 From Numbers to Bits(2)

- **Integer Formats(2)**
  - Binary words to represent the integers using ***scientific notation***
  - For example, consider the integer 62,000,000,000,000,000;
    - ✓ Very long word length (56 bits,as it turns out)
    - ✓ Scientific notation to write this number as 62×10^{15}
    - ✓ The two numbers 62 and 15 to represent the large integer.
    - ✓ The numbers 62 and 15 could be stored using 6 and 4 bits respectively, for a total of 10 bits.
- **BCD Formats**
  - ✓ The BCD codes

Numeral	BCD Representation
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

- For example, we can represent 749 as:

0111 0100 1001,

3.4 Bits as Building Blocks of Information(9)

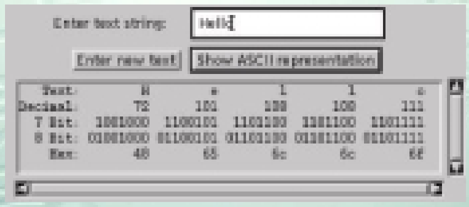
3.4. 4 Representing Text with Bits

- **ASCII**, pronounced ``ask-key," is an acronym for ***American standard Code for Information Interchange***. ASCII is used to encode text, and in particular is useful for representing information which is entered via a computer keyboard. Therefore, ASCII must be able to represent:
  - numerals;
  - letters in both upper and lower cases;
  - special ``printing symbols such as @, $, *, &, and %; and
  - commands that are commonly used by computers to represent carriage returns, line feeds, and other text-formatting directives.
- A complete list of ASCII characters is given in Appendix A.
- Ex) "You & I," would be represented in ASCII by the 56-bit sequence: → 1011001 1101111 1110101 0100000 0100110 0100000 1001001 0101100,

3.4 Bits as Building Blocks of Information(10)

3.4. 4 Representing Text with Bits(2)

- *Alphanumeric String to ASCII Conversion Applet*



3.4 Bits as Building Blocks of Information(11)

3.4. 5 Binary Representation of More Complex Information

- **Representing Real Numbers--Precision and Accuracy**
  - Can we use bits to represent more general types of information?
  - The answer to this question is ***yes***.
- **Real Numbers**
  - Can determine the precision with which we will represent the temperature by using the appropriate number of bits.
  - An 8-bit word can take on $2^8 = 256$ different values, and a 16-bit word can take on $2^{16} = 65{,}536$ different values. So, if we use an 8-bit word to represent this temperature, we can represent 256 different temperatures between 60 and 70 degrees.
  - **Floating-point representation**

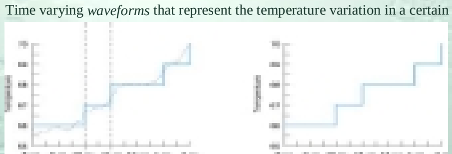
Binary Codeword	Temperature, ° F
0000 0000	60.000
0000 0001	60.039
0000 0010	60.078
0000 0011	60.117
.	.
.	.
.	.
1111 1101	69.883
1111 1110	69.922
1111 1111	69.961

Information Technology 17

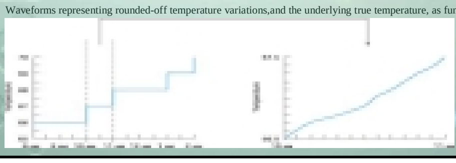
3.4 Bits as Building Blocks of Information(12)

3.4. 5 Binary Representation of More Complex Information(2)

- **Functions of Time**
  - Capture, store, process, or transmit is time-varying information; many information sources vary as a *function of time*.
  - Time varying *waveforms* that represent the temperature variation in a certain place over



  - Waveforms representing rounded-off temperature variations,and the underlying true temperature, as functions of time.



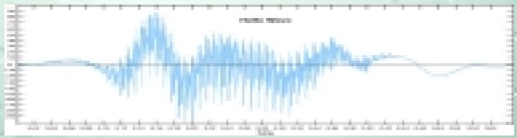
18

6

3.4 Bits as Building Blocks of Information(13)

3.4. 5 Binary Representation of More Complex Information(3)

- **Functions of Time (2)**
  - An audio *waveform* that represents the modulation of air pressure that took place when a person spoke the word ``hello.''



  - For a CD, 16 bits are used for each value we represent.
  - Generating 16 binary digits every 0.0000227 seconds, or, by generating 16 bits 44,100 times per second.

$$16 \text{ bits/value} \times 44{,}100 \text{ values} \times 2 \text{ channels} = 1{,}411{,}200 \text{ bits}$$

- **Still and Time-Varying Images**

3. 5 Convenient Forms for Binary Codes

3.5.1 Bits, Bytes, and Beyond

- Prefix
  - ``K'' is used to represent 2^{10} = 1024, and
  - ``M'' is used to represent 2^{20} = 1,048,576, or .
  - This is done so that these prefixes can represent powers of two; as a result, the number of KB (1,024 bits) or MB(1,048,576 bits) is usually a convenient number, easy to remember.
- Octal

Octal Numeral	Bit Pattern
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

  - 12-bit pattern is stored in a computer's memory: 010110011101_2,
  - we can represent these bits in octal as: 2635_8.

3. 5 Convenient Forms for Binary Codes (2)

3.5.1 Bits, Bytes, and Beyond

- Hexadecimal
  - The hex system is a counting system that uses 16 numerals

Decimal	Octal	Hex Numeral	Bit Pattern
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	10	8	1000
9	11	9	1001
10	12	A	1010
11	13	B	1011
12	14	C	1100
13	15	D	1101
14	16	E	1110
15	17	F	1111

  - ASCII → Hexa
    “You & I,” = 1011001 1101111 1110101 0100000 0100110 0100000 1001001 0101100.
    = 5F 6F 75 20 26 20 49 2C
  - **Binary / Decimal / Octal / Hexadecimal Conversion ???**

3. 5 Convenient Forms for Binary Codes (3)

3.5.3 Introduction to Error Detection and Correction

- One important function of **coding** is to enable the ***detection*** (and often ***correction***) **of errors** that can occur in data transmission across a noisy channel. In fact, this ability to detect and correct errors is one of the primary advantages of using digital transmission instead of analog.
- As an example, consider one of the simplest forms of error detection, data coding using a one-bit *parity code*.
- **Table 3.1:**Even Parity Example

Transmitted Character	Transmitted Information	Transmitted Parity	Received Information	Received Parity	Do we Detect an Error?
H	1001000	0	1000000	0	Yes
e	1100101	0	1100101	0	No
l	1101100	0	1101100	0	No
p	1110000	1	1110000	1	No