Data Representation and Problem Solving

Instructor: Dmitri A. Gusev

Fall 2007

CS 113: Introduction to Computers

Lecture 2, September 6, 2007
Bits, Bytes, Words...

1 bit is a storage unit that must contain either a 0 or a 1.

1 byte is a unit consisting of 8 bits.

A word is a group of one or more bytes.

(Pentium 4 is a 32-bit machine, 4 bytes per word.)
Data Representation

Analog (continuous) vs. Digital (discrete)
Signed-Magnitude Representation of Negative Numbers

Add an extra bit on the left to represent the sign. Use 0 for the ‘+’ sign, 1 for the ‘-’ sign.

*Example* (3 bits allocated for the magnitude, 1 bit for the sign):

0101 = 5_{10}, 1101 = -5_{10}.

Problems with the *signed-magnitude* representation:

- Two representations of 0: 0000 and 1000;
- Special logic is required to perform addition, subtraction, multiplication and division.
Ten’s Complement Representation of Negative Numbers

- Limit the maximum number of decimal digits by $k$.
- Interpret the first half of numbers ($0, 1, \ldots, (10^k/2)-1$) as natural numbers. Interpret the other numbers as

  $$\text{Negative}(m) = 10^k - m$$

- Example, $k=3$: $123 + (-455) = 123 + (1000-455) = 668_{10c} = -332_{10}$
Two’s Complement Representation of Negative Numbers

- Representing

\[
\text{Negative}(m) = 2^k - m, \\
\text{where } k \text{ is the number of bits used.}
\]

Example: \( k=8 \),

\[
\begin{array}{c|c|c}
\text{-125} & 10000011 & (256-125=131=1+2+128) \\
+ 3 & \phantom{0}00000011 & \\
\hline
\text{-122} & 10000110 & \text{“134”}=\text{Negative}(122))
\end{array}
\]

Overflow will occur if the result of addition exceeds 127: “128” (100000000) now serves as (-128)!
Representing Real Numbers

- **Scientific notation:**
  
  \[.00508259 = 5.08259 \times 10^{-3} \rightarrow 5.08259E-3\]
  
  The *decimal point* is kept to the right of the most significant (non-zero) digit.

- **Floating point:** A real value in Base 10
  
  \[r = \text{sign} \times \text{mantissa} \times 10^{\text{exponent}}\]
  
  The # of digits is fixed, but the point “floats”.

- In other bases, the analog of the decimal point is called a *radix point*. 
Representing Real Numbers in Binary

\[ r = \text{sign} \times \text{mantissa} \times 2^{\text{exponent}} \]

How to convert the fractional part from decimal to binary? Keep multiplying by the base and reading off the digits. Example:

\[17.875_{10} = 10001.111_2\]

\[17/2=8.5, \ 0.5 \times 2=1; \ 8/2=4; \ 4/2=2; 2/2=1; \]

\[.875 \times 2=1.75; \ 0.75 \times 2=1.5; \ 0.5 \times 2=1.\]
Representing Text

• Encoding characters vs. *formatting* (fonts, margins, tables, color, etc.)

• A *character set* is a list of characters and the codes used to represent them. How many characters do we need?..

The Unicode Character Set

- 16 bits per character. $2^{16}=65536$ unique characters can be represented.
- The first 256 characters in the Unicode set correspond to those of the extended ASCII character set. ("Backward compatibility").
Program Development Cycle

1. **Analyze**: What should the output be? What data/input is necessary to obtain the output?

2. **Design**: Develop an algorithm – a logical sequence of precise steps that solve the problem

3. **Choose the interface**: Create command buttons and menus to allow the user to control the program

4. **Code**: Translate the algorithm into a programming language

5. **Test and debug**: Locate and remove any errors (“bugs”) in the program

6. **Complete the documentation**: For commercial programs, develop an instruction manual and on-line help
Flowchart Symbols

- Flowline
- Terminal
- Input/Output
- Processing
- Decision
- Connector
- Offpage Connector
- Predefined Process
- Annotation
Sample Flowchart: Factorial
Pseudocode

_Pseudocode_ is a compact and informal high-level description of a computer programming algorithm that uses the structural conventions of programming languages, but omits detailed subroutines, variable declarations or language-specific syntax. The programming language is augmented with natural language descriptions of the details, where convenient. Pseudocode generally does not actually obey the syntax rules of any particular language; there is no systematic standard form, although any particular writer will generally borrow the appearance of a particular language.