CS1101: Lecture 30

ISA & Floating-Point Numbers

Dr. Barry O'Sullivan
b.osullivan@cs.ucc.ie

Course Homepage
http://www.cs.ucc.ie/~osullb/cs1101

Department of Computer Science, University College Cork

Lecture Outline

- Introduction to Floating Point
- Principles of Floating Point
- An Example Representation
- Modelling the Real-Number System
- Density & Errors
- Rounding
- Effects of Changes in the System
- Relative Error
- Normalization
- **Reading**: Tanenbaum, Appendix B.

Introduction

- In many calculations the range of numbers used is very large.
- For example:
  - the mass of the electron, 9×10^{-28} grams,
  - mass of the sun, 2×10^{33} grams,
- This range exceeds 10^{60}!!!
- On a binary computer, multiple-precision arithmetic could be used to provide enough significance (we don't do this though).

How do we handle this?

- However, few measurements of any kind can (or need) be made accurately to 62 significant digits.
- Although it would be possible to keep all intermediate results to 62 significant digits and then throw away 50 or 60 of them before printing the final results
- This is wasteful of both CPU time and memory.
- What is needed is a system for representing numbers in which the range of expressible numbers is independent of the number of significant digits.
- We will consider such a system based on scientific notation.

Principles of Floating Point

- In the scientific notation $n = f \times 10^e$

  where f is called the **fraction**, or **mantissa**, and e is a positive or negative integer called the **exponent**.

- The computer version of this notation is called floating point.

- Examples
  - $3.14 = 0.314 \times 10^1 = 3.1410 \times 10^0$
  - $0.000001 = 0.1 \times 10^{-5} = 1.0 \times 10^{-6}$
  - $1941 = 0.1941 \times 10^4 = 1.941 \times 10^3$

- The *range* is effectively determined by the number of digits in the exponent.

- The *precision* is determined by the number of digits in the fraction.

An Example Representation

- Because there is more than one way to represent a given number, one form is usually chosen as the standard.

- Consider a representation, R, with a signed three-digit fraction in the range $0.1 \leq |f| < 1$ or zero and a signed two-digit exponent.

- These numbers range in magnitude from $+0.100 \times 10^{-99}$ to $+0.999 \times 10^{+99}$, a span of nearly 199 orders of magnitude, yet only five digits and two signs are needed to store a number.

Modelling the Real-Number System

- Floating-point numbers can be used to model the real-number system of mathematics, although there are some important differences.

- The real line is divided up into seven regions:
  1. Large negative numbers less than -0.999×10^{99}.
  2. Negative numbers between -0.999×10^{99} and -0.100×10^{-99}.
  3. Small negative numbers with magnitudes less than -0.100×10^{-99}.
  4. Zero.
  5. Small positive numbers with magnitudes less than 0.100×10^{-99}.
  6. Positive numbers between 0.100×10^{-99} and 0.999×10^{99}.
  7. Large positive numbers greater than 0.999×10^{99}.

Modelling the Real-Number System

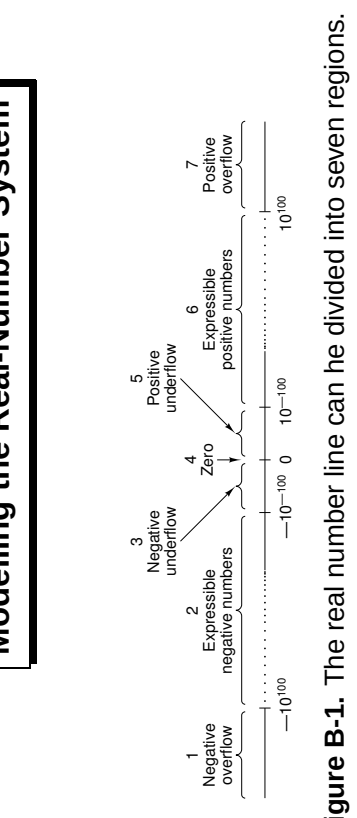

Figure B-1. The real number line can he divided into seven regions.

Errors

- Unlike real numbers, the set of numbers representable with three fraction and two exponent digits cannot be used to express any numbers in regions 1, 3, 5, or 7.
- A number in regions 1 or 7, means an **overflow error** has occurred.
- A number in region 3 or 5 – **underfiow error**.
- Underflow error is less serious than overflow error, because 0 is often a satisfactory approximation to numbers in regions 3 and 5.

Density

- Unlike real numbers, floating-point numbers do not form a continuum.
- Exactly 179,100 positive numbers can be expressed in the five-digit, two-sign system used above, 179,100 negative numbers and 0 for a total of 358,201 numbers.
- Of the infinite number of real numbers between -10^{+100} and $+0.999 \times 10^{99}$, only 358,201 of them can be specified by this notation.

Rounding

- It is quite possible for the result of a calculation to be one of the other numbers, even though it is in region 2 or 6.
- For example, $+0.100 \times 10^3$ divided by 3 cannot he expressed exactly.
- The obvious thing to do is to use the nearest number that can be expressed.
- This process is called **rounding**.

Relative Error

- The "space" between adjacent expressible numbers in regions 2 and 6 is not constant.
- The separation between $+0.998 \times 10^{99}$ and $+0.999 \times 10^{99}$ is very different than that between $+0.998 \times 10^0$ and $+0.999 \times 10^0$
- However, when separation between a number and its successor is expressed as a percentage of that number, there is no systematic variation throughout region 2 or 6.
- The **relative error** introduced by rounding is approximately the same for small numbers as large numbers.

Effects of Changes in the System

- Changing the number of digits in the fraction or exponent shifts the boundaries of regions 2 and 6 and changes the number of expressible points in them.
- Increasing the number of digits in the fraction increases the density of points and therefore improves the accuracy of approximations.
- Increasing the number of digits in the exponent increases the size of regions 2 and 6 by shrinking regions 1, 3, 5, and 7.

Normalization

- A variation of this representation is used in computers.
- For efficiency, exponentiation is to base 2, 4, 8, or 16 rather than 10, in which case the fraction consists of a string of binary, base-4, octal, or hexadecimal digits.
- If the leftmost of these digits is zero, all the digits can be shifted one place to the left and the exponent decreased by 1, without changing the value of the number (barring underflow).
- A fraction with a nonzero leftmost digit is said to be **normalized**.
- Normalized numbers are generally preferable to unnormalized numbers, because there is only one normalized form, whereas there are many unnormalized forms.