Exercises for Lectures by Kurt Mehlhorn

Kurt Mehlhorn

July 17, 2008

**Exercise 1** Let $S$ be a set of double precision floating point numbers.

- We want to sort them and use the following implementation of the compare routine:

  ```c
  int compare(double x, double y)
  { if (x > y) return +1;
    if (x = y) return 0;
    if (x < y) return -1;
  }
  ```

  Here $>$, $=$, and $<$ are the built-in comparison operators. Will this sort correctly?

- We want sort the numbers according to the square of their value and use the following implementation of compare.

  ```c
  int square_cpm(double x, double y) { return compare(x * x, y * y); }
  ```

  Is this correct? If not, design a comparison function that will work with floating point arithmetic.

**Exercise 2** Let $A$ be an $m \times n$ matrix and let $b$ be an $m$ vector. We are interested in the system $Ax = b$ of linear equations. For this exercise, we assume exact real arithmetic.

- Assume you have an $m$ vector $c$ such that $c^T A = 0$ and $c^T b \neq 0$. Show that this proves that the linear system $Ax = b$, where $x$ is an $n$ vector, has no solution.

  For example, the system

  
  
  $\begin{align*}
  2x + 3y &= 7 \\
  2x + 3y &= 8
  \end{align*}$

  has no solution because one times the first equation minus one times the second equation yields the equation $0x + 0y = -1$, which is clearly unsolvable.
• Prove that such a $c$ always exists, if $Ax = b$ is not solvable.
  
  Hint: Gaussian elimination can be used to solve linear systems. What can you infer when Gaussian elimination fails to solve a system.

• Design a certifying algorithm for solving linear systems. What does it return when the system is solvable, what does it return when the system is unsolvable?

**Exercise 3** Recapitulate the programs that you have written in the past 12 months. Would you write the program differently had you known about the concept of a certifying algorithms? If yes, why? If no, why?

**Exercise 4** Write a program to find convex hulls of points in the plane. Implement it with floating point arithmetic. Find inputs for which your program does not work.