Effective Computational Geometry
Exercise 4

Lutz Kettner Kurt Mehlhorn Elmar Schömer

November 22, 2001

Exercise 1: Orientation of Circle Points: Recall that a circle point is the intersection of a rational circle and a rational line. In class we derived formulae for the x-coordinates of circle points. We can get the y-coordinates by plugging the x-coordinates into the line equation defining the point. Assume now that we have three circle points p, q, and r (each defined by its own circle and line). Also recall that the orientation predicate is given by

$$\mathit{sign}\,\det\begin{pmatrix} 1 & 1 & 1 \\ p_x & q_x & r_x \\ p_y & q_y & r_y \end{pmatrix}$$

1. (5 pts) How many distinct square roots occur?
2. (5 pts) How many rounds of squaring are needed in the repeated squaring approach to sign determination?
3. (5 pts) Show how to compute the sign by repeated squaring. You may stop the process after writing one page. The purpose of the exercise is to give you personal experience with the repeated squaring strategy.
4. (10 pts) In the separation bound approach, there are quantities $k(E)$ and $u(E)$. How large is $k(E)$? Give an upper bound on $u(E)$ under the assumption that the line coefficients, the squared radii, and the coordinates of the circle centers are L-bit integers.

Exercise 2: Quality of Separation Bound: (10 pts) Consider the number E obtained in the following way. Square the number 2 k times, add 1, take square roots k times and subtract 2. E is non-zero, but close to zero. Use the separation bound shown in class to derive a lower bound on the value of E. Call this lower bound L. Derive an upper bound U on the value of E and estimate $(\log U)/(\log L)$. Try to show that this quotient is bounded by a constant independent of k. Hint: Use the estimate $\sqrt{1+x} \leq 1+x$ for $x \geq 0$.

Exercise 3: Arbitrary Precision Floating Point Arithmetic: Let L be a fixed positive integer. For this exercise a floating point number is a triple (s, m, e) where $s \in \{-1, 0, +1\}$, m is an non-negative integer with $0 \leq m < 2^L$ and e is an arbitrary integer. The triple represents the real number $s \cdot m \cdot 2^e$. The main difference to standard floating point arithmetic is that the exponent range is unbounded and hence there is no over- and underflow.

1. (5 pts) Design algorithms for addition and multiplication which minimize the round-off error. Of course, you may presume algorithms for the addition and multiplication of integers.

2. (5 pts) We now add a range of uncertainty, i.e., we consider quadruples of the form (s, m, e, B) where B is a non-negative integer. Such a quadruple represents all numbers of the form $(s, m \pm i, e)$ with $0 \leq i \leq B$. Design addition and multiplication formulae.

3. (5 pts) In the scheme discussed in the preceding item, B could become a number of arbitrary length. Assume now that $L = 32k$ and that we want to restrict B to a 64 bit integer. Whenever, B becomes larger than 2^{64}, we write $m = m_1 2^{32} + m_2$ with $m_2 < 2^{32}$ and $B = B_1 2^{32} + B_2$ and then replace the triple (s, m, e, B) by the triple $(s, m_1, e+32, B')$, where $B' =$ that's for you to fill in. In this way, we need a bounded number of bits for the range of uncertainty and numbers with large uncertaintly have short mantissae (i.e., we do not keep bits that are unreliably anyway). The scheme just described is called *ranged arithmetic*.

Solutions are due November 29th and will be collected in the lecture on November 29th. Solutions will be discussed in the tutorials of the week after.