06-35324 *Mathematical and Logical Foundations of Computer Science* University of Birmingham
Autumn Semester 2021 School of Computer Science
© Miriam Backens October 4, 2021

Exercise Sheet 2 - Mathematics

Unassessed exercises

Write out your answers to all exercises and submit via Canvas by next week, Tuesday, 11am. (We will review a sample of answers but not be able to give feedback to everyone.)

Exercise 2.1

Compute the complete value table for the polynomial

$$p(x) = 2x^2 + 3x + 1$$

in the field GF(5).

(By "value table" I mean a table that gives $p(x)$ for all elements x in GF(5).)

Exercise 2.2

Prove that

$$x = u_x \times a + v_x \times b \qquad \text{and} \qquad y = u_y \times a + v_y \times b\ .$$

is an invariant for the extended Euclidean algorithm (as claimed in Section 3.4 of the course booklet).

Exercise 2.3

Solve the linear equation $4x = 3$ in the finite field $\mathbb{Z}_{17}$.

Exercise 2.4

There is a field with four elements (called GF(4)) but its addition and multiplication are different from those of $\mathbb{Z}_4$. Let's call its elements 0, 1, △ and □. Below are the addition and multiplication table for this field partially filled in. Use the field laws to complete them.

+	0	1	△	□
0				
1		0	□	
△			0	
□				

×	0	1	△	□
0				
1				
△			□	
□				

Exercise 2.5

Find the 32-bit floating point number that is closest to the integer 123456789.

(Hint: Translate 12456789 into binary first.)

Exercise 2.6 — just for fun

All operating systems come with a built-in (pseudo) random number generator, which for simplicity we assume produces random bits, that is, it produces 0 or 1 with equal probability $1/2$. Java uses this facility to provide a function `Math.random()` which returns random 64-bit floating point numbers in the interval $[0,1]$. This function has the following properties:

(a) Every 64-bit floating point number between 0 and 1 (inclusive) has a non-zero probability of being returned by `Math.random()`.

(b) For every $0 \leq a < b \leq 1$, the probability of receiving a result in $[a,b]$ is very close to $b-a$. (I say "very close", because by the discrete nature of the floating point numbers it is impossible to get a perfect uniform distribution.)

This is remarkable, because we know that the floating point numbers are not evenly distributed. How does Java do it? Or, how would you do it?

If you have any good ideas, then send them directly to me at `m.backens@cs.bham.ac.uk`