Mathematical and Logical Foundations of Computer Science — Summary of Lecture 3 —

Achim Jung
School of Computer Science
University of Birmingham, UK

Autumn 2020

Fractions and rational numbers

- Fractions are added and multiplied as we learned in school but we discovered that the laws of arithmetic (the distributivity law in particular) are only valid if we identify fractions that express the same ratio, so there is a difference between "fractions" and "rational numbers". It's the rational numbers we want. They are denoted by $\mathbb{Q}$.

- The rational numbers do satisfy all the ring laws and they also allow us to form a multiplicative inverse (and hence division), so we say that they are not only a ring but a field.

- Multiplicative cancellation is valid in every field, so we no longer need to state it separately.

Reduced fractions

- Instead of considering *all fractions* as valid, we can restrict attention to those fractions where denominator and numerator have no common factor (> 1). We call those “reduced”.

- In order to find the largest common factor we use Euclid’s algorithm. It’s very efficient.

- We showed the correctness of the algorithm by a technique called loop invariants. Showing the loop invariant to hold is very reminiscent of a proof by induction.

Finite fields

- Doing careful bookkeeping during Euclid's algorithm allows us to see that Bézout's Identity holds:

$$\mathsf{lcf}(a, b) = u \times a + v \times b$$

- From this it's a simple step to realise that if m is a prime number, then every non-zero element of $\mathbb{Z}_m$ has a multiplicative inverse, so these $\mathbb{Z}_m$ are actually fields and not just rings.

- The multiplicative inverse of a number in $\mathbb{Q}$ (the rationals) is **very different** from its multiplicative inverse in such a $\mathbb{Z}_m$. Example:

$$\mathbb{Q} : 2^{-1} = \frac{1}{2} \qquad\qquad \mathbb{Z}_5 : 2^{-1} = 3$$

On a computer...

- Rational numbers can easily be implemented but this is hardly ever done (because numerators are growing very fast during arithmetic operations).

- Finite fields, on the other hand, are very prominent in cryptography and error-correcting codes.