Mathematical and Logical Foundations of Computer Science — Summary of Lecture 2 —

Achim Jung
School of Computer Science
University of Birmingham, UK

Autumn 2020

Integers — the highlights

- We only need to assume that for every integer a there is another integer denoted by $-a$ for which $a + (-a) = 0$ holds.
- From this we can prove (additive) cancellation.
- From this we can prove annihilation.
- From this we can prove double negation: $-(-a) = a$.
- From this we can prove minus times minus equals plus: $(-a) \times (-b) = a \times b$.

The situation is common in mathematics and computer science and is called a ring. We say, "the integers form a ring".

Multiplicative cancellation also holds for the integers but in other rings it may fail.

Computer integers

- Java's `int` variables are based on 32-bit registers.
- All calculations are done modulo 2^{32}.
- The bit patterns from $100\ldots000$ to $111\ldots111$ are interpreted as negative numbers.

General modulo arithmetic

- Computing "modulo m" can be done for any $m > 1$. We get the ring $\mathbb{Z}_m$ which has exactly m different elements.

- Calculations in $\mathbb{Z}_m$ can be thought of in two different ways:

  1. We can take the numbers from 0 to $m - 1$ as the standard members of $\mathbb{Z}_m$, perform calculations with them as we would in $\mathbb{Z}$, then reduce the result to an answer between 0 and $m - 1$ at the end. Example in $\mathbb{Z}_7$

     $$\begin{array}{rcll} 3 \times 5 & = & 15 & \text{in } \mathbb{Z} \\ & \equiv & 1 & \text{modulo } 7 \end{array}$$

     So in $\mathbb{Z}_7$ we have $3 \times 5 = 1$.
  2. Alternatively, we can do all calculations in $\mathbb{Z}$ and but use $\equiv$ for comparisons, instead of $=$.

- Computer integers implement calculations in $\mathbb{Z}_{2^{32}}$ and adopt the first approach internally, but when reporting the result back to the user, the numbers between 2^{31} and $2^{32} - 1$ are converted to negative numbers by subtracting 2^{32}.