Concepts of Programming Languages
Lecture Notes: Undefined Behavior and Safety

Stefan Mitsch

School of Computing, DePaul University
smitsch@depaul.edu

Learning Goals

🎓 Identify how C treats corner cases by leaving them "undefined" and how this leads to *unsafe* programs

Undefined Behavior

Many mathematical functions $f : A \to B$ are not defined over their entire domain (remember that we call the values of the set A a function's domain, and the values of the set B its range or co-domain). For example, division $\text{div} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a function that takes two real values and returns one real value, but division by zero is undefined. When we specify division as an operation in a computer, however, we need to specify how it behaves on all inputs from the domain. There are several ways of achieving this.

Exercise 1 (Language constructs for undefined behavior). Before you read on, can you think of some ways of specifying a function such that it avoids undefined behavior?

- Define a type $\mathbb{R}_{\neq 0} \equiv \mathbb{R} \setminus \{0\}$ and define division as $\text{div} : \mathbb{R} \times \mathbb{R}_{\neq 0} \to \mathbb{R}$
- Check arguments in the function definition and use exceptions to report violation
- Make it the responsibility of the user to not misuse the function (i.e., the C approach, just return any value on division by zero)
- Define division by zero to be zero (the approach in the theorem prover Isabelle)

This example is also a good reminder that numerical computations with mathematical sets of infinite size, such as $\mathbb{R}$, are challenging to be represented in a computer with finite memory and finite bitsize representations (symbolically, we can represent and reason about the properties of infinite sets and operations on infinite sets, with finite memory we just cannot represent every element of an infinite set with its own symbol). Specific programming languages may have other sources of undefined behavior. For example, in C, over- and underflow of variables, evaluation sequence in expressions, and dangling pointers can result in undefined behavior.