Axi Design: Language Features

Wojciech Kołowski

What is a function, classically?

In classical mathematics, a relation R between A and B is called a function when it is:

- Right-unique (also called deterministic): for all $a : A$ and $b_1, b_2 : B$, if $R(a, b_1)$ and $R(a, b_2)$, then $b_1 = b_2$.
- Right-total (also called total): for all $a : A$, there exists $b : B$ such that $R(a, b)$.

What is a function, constructively?

In constructive mathematics, TODO

What is totality checking?

Totality checking is the following task: we are given the definition of a well-typed function $f : A \to B$ and we need to tell whether it is total, i.e. whether $\forall a : A, \exists b : B, f(a) = b$. Spelled out in words: for all a of type A, there exists b of type B such that $f(a)$ equals b.

Why should we care?

Why should we care about totality checking? TODO

Why should we care?

We already know that all functions must terminate for proof checking to be decidable. However, this isn't termination checking's raison d'être.

Neither has it anything to do with gas, or even what happens at runtime at all. After all, the universe is going to last a finite amount of time, so all functions will eventually terminate, right? Wrong.

Termination is first and foremost a logical notion. It's evil twin, non-termination, manifests itself most strikingly not at runtime, where it cannot be observed at all (because it would take forever), but in the logic – TODO

Termination checking is undecidable

In general, termination checking is undecidable – it means literally solving the Halting Problem! That's hard!