

# — Sprint 2 —

## Deadline: 8th of May

Thanks for your first prototype! Our engineers will now deploy it to our BrAvE<sup>1</sup> customers and collect some real-world experience. Meanwhile our RLD<sup>2</sup> department which specializes in the analysis of disruptive technologies has identified some additional use cases in WolVeriNe. They require two extensions to your solver:

## Multiple sorts

Until now all variables were of the same type (*sort*). To improve readability (and offload checking the type safety of our system to your solver) we now want to properly annotate the variables with the correct types.

Your framework should already support parsing this and should also enforce type safety on the parsed formula. However the formula you need to check contains different variable types now.

## Function symbols

Our own verification team turns out to be pretty lazy. Therefore there is a pretty large set of stuff that we can not properly encode into an SMT problem (yet). We decided to solve this problem by abstracting from the details of this stuff and simply replace it by some function. Of course our analysis might miss certain cases... but whatever. We just pretend that if some implementation could exist that meets the specification that this is what the implementation does. The customers won't notice it anyway...

We will specify certain function symbols and you can essentially make the function do whatever you want. The only restriction is that it is deterministic – for a certain set of input values every function always returns the same output. Or formally (for a binary function  $f$ ):

$$\forall a, b, c, d : (a = c \wedge b = d) \Rightarrow (f(a, b) = f(c, d))$$

If such a function exists, please tell us how it should look like by listing what it returns for every input that is relevant. Please check the Slack channel general for some examples.

---

<sup>1</sup>Broken software for the AdVEnturous

<sup>2</sup>Ridiculously Large Data