1 Testing

The previous section showed that a rigorous definition of fixed-point arithmetic suitable for use in writing standards can be written. That is not surprising. The claim of this paper is that machine-checkable specifications *should* be written for appropriate aspects of standards, because machine checking is *needed*.

We can classify errors in a specification as

- *Syntax* errors, such as incorrect punctuation, easily caught by a parser. Reducing these calls for a notation that has a checkable grammar.
- *Type* errors, such as calling the wrong function, or providing the wrong number of arguments, or swapping two arguments, caught by a more or less capable type checker. Reducing these calls for a notation that has a statically checkable type system.
- Other *Semantic* errors, such as swapping two arguments of the same type, or sign errors. Reducing these requires a notation that can be executed in order to run tests, and ideally a testing framework that makes it easy to write tests.

The specification in the previous section is small, but during its development there were numerous errors of each kind.