The quest for correctness

Henk Barendregt

Nijmegen University

1. The problem and its solution

How to obtain reliability? By proofs

2. Representing proofs

By types

1. The problem

Truth

Scientific interest

Industrial interest: product quality

product satisfies expectation (client)

design meets specification (engineer)

Products

Wheel	Bridge
Hardware	Software

A wheel can be seen to be correct

A bridge can be tested to be correct

For hardware and software this is impossible

The correctness of hardware is concerned with an over-astronomical number of states

The correctness of software is concerned with a virtually infinite number of states

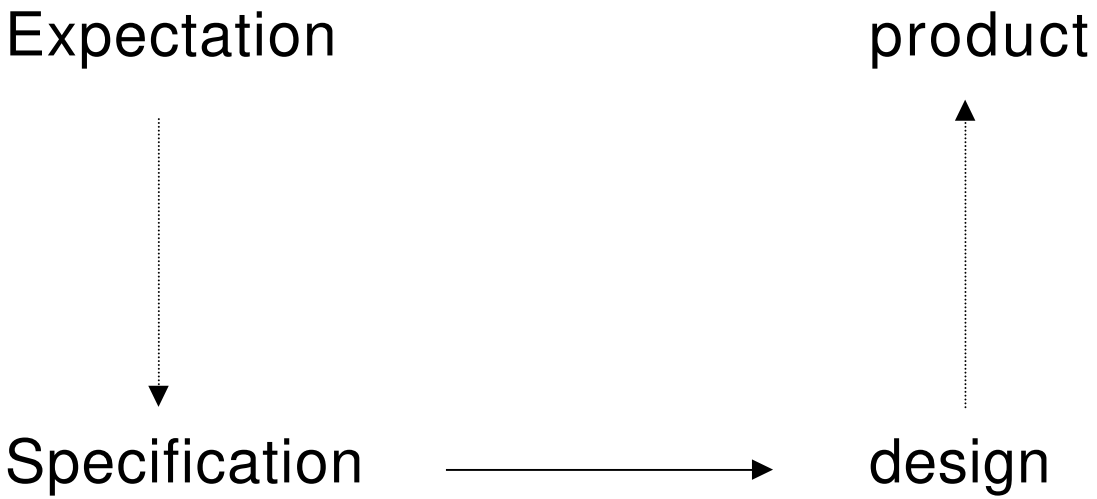
We need proofs of correctness for complex statements

$\Rightarrow$

proofs for the reliability of products

How can it be that there is a relation between

a formal proof and the reliability of a product?



H. Wupper

The dashed arrows have an empirical component and should be beyond doubt

The hard work is in the design (from the spec) and the validation (testing, proving) of it

But specifying is also an art; it is common to do it wrong

When train passes sensor S1 or T1 gate closes;
When train passes sensor S2 or T2 gate opens

This is wrong: a train may have come from the other side

in the meantime

Better

There is never a train when the gate is open

More formally

For all $t \in R$ [Open(t) $\Rightarrow$ Train(t)]

This does satisfy the expectation of the client, but cannot be proved

$$\text{Assumption} \Rightarrow \text{for all } t \in R\,[\,\text{Open}(t) \Rightarrow \text{Train}(t)\,]$$

where

assumption =

Train(t) $\Rightarrow$ “train has passed sensor before”

Conclusion: we need a sophisticated language

The language of mathematics is rich enough

Why are proved statements reliable?

Because they are derived step by step

from the assumptions using logic

The few allowed steps in logic are seen to be correct

How can we be sure that a statement that is claimed to be proved is really proved?

By asking for a proof and verify all the steps

This can be mechanised

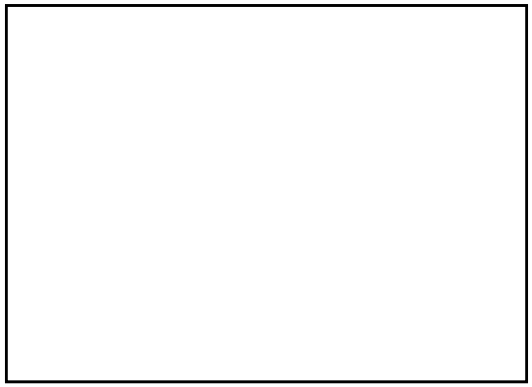
By asking for a fully formalised proof and verify by computer all the steps

The verifying program should be simple (de Bruijn criterion)

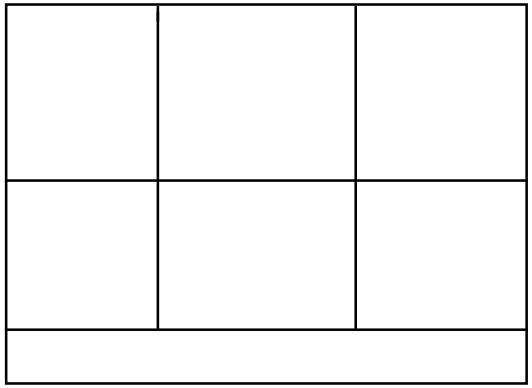
This is better than requiring that a proof should be produced by computer

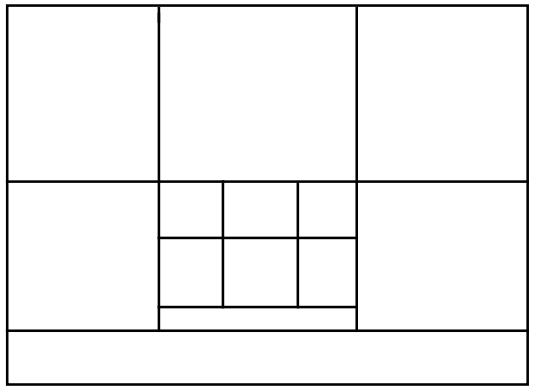
Complex products consisting of components

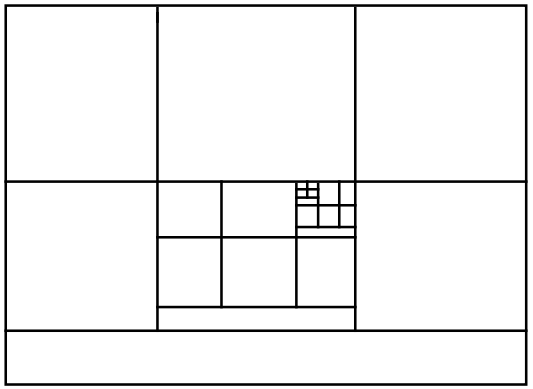
consisting of components ....

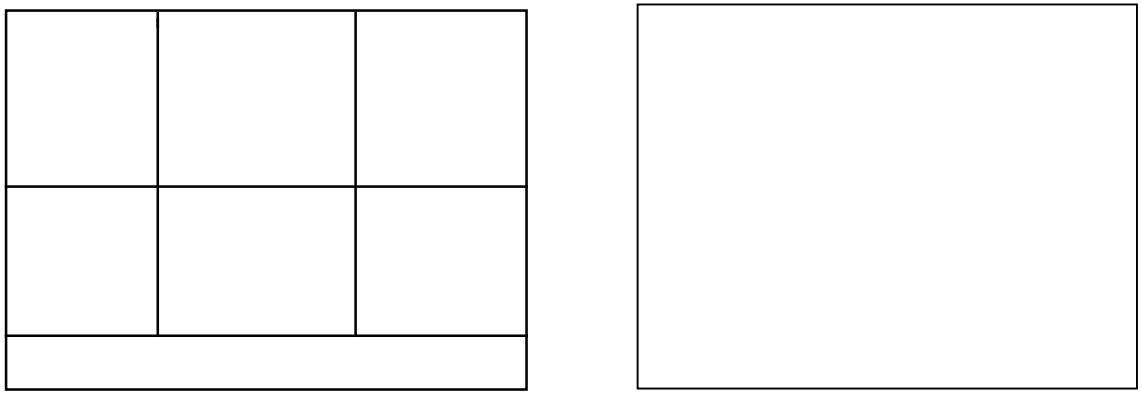


The Chinese box









$S_1(d_1)$ & … & $S_6(d_6)$ $\Rightarrow$ $S(d)$ $[d= f(d_1,\ldots,d_6)]$

The methodology of design with formal verification has been applied *in some sense* to the construction of hardware

In principle it can be applied also for the correctness of mathematics and software