2 function definitions for dummies.

The syntax is rather self-explanatory: We introduce a function by giving its name, its type, and a set of defining recursive equations. If we leave out the type, the most general type will be inferred, which can sometimes lead to surprises: Since both 1 and + are overloaded, we would end up with fib :: nat ⇒ 'a::{one,plus}. Recursive calls which are nested in one another frequently cause complications, since their termination proof can depend on a partial correctness property of the function itself. As a small example, we define the "nested zero" function: function nz :: "nat ⇒ nat" where...