So far there are 3 elements that are introduced for determining type signatures

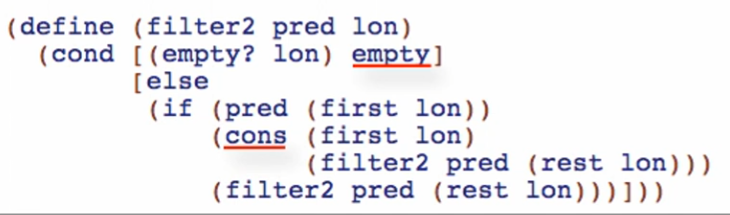
* (listOf X) instead of ListOfX
* When a function is passed as an argument, we write its signature in parentheses in the larger signature
* Use of type parameters (single upper-case letters) to make sure types are consistent where they need to be

Working through filter2 function

Initial signature



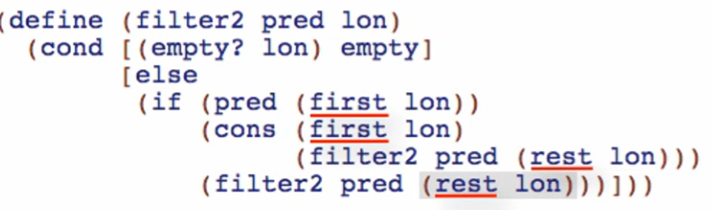
Checking the type produced



Empty and cons = clearly a listof something



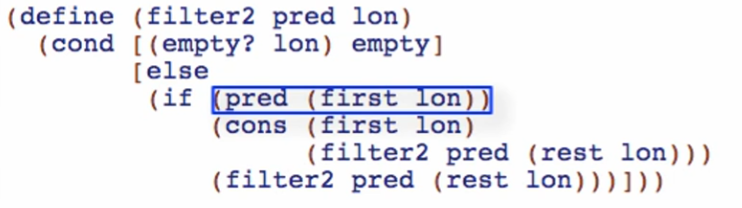
Checking the “lon” parameter



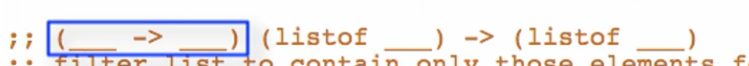
First and rest = clearly a list of something!



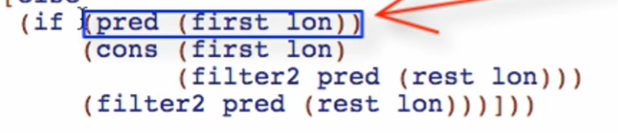
Checking the “pred” parameter



pred here is a one-argument function



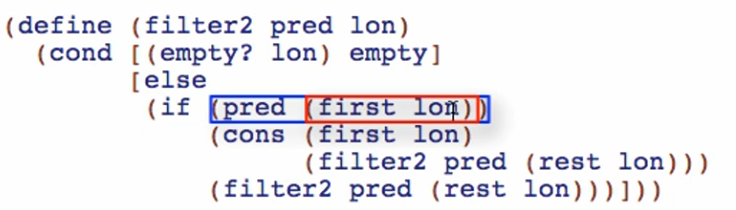
Notice pred is used as a QUESTION for an if statement



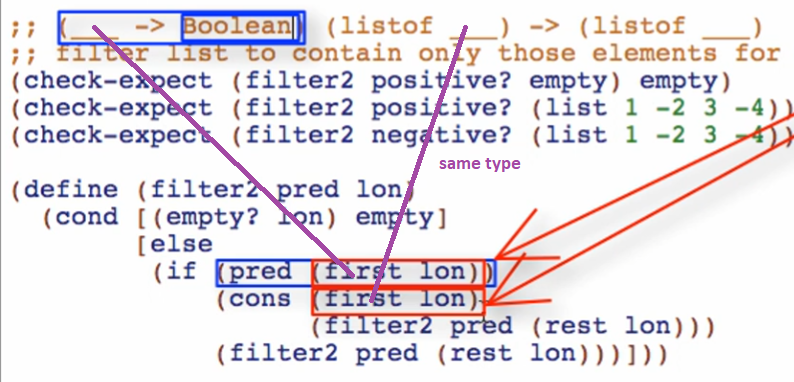
Therefore, it should produce true or false, in other words, it should produce a Boolean



Checking the argument to pred



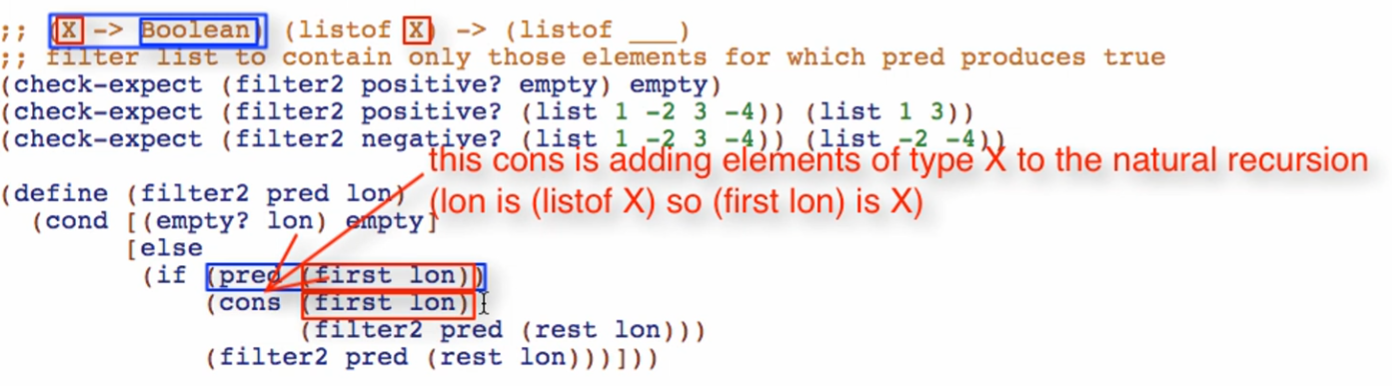
Again, filter2 does not look at the first lon. Only cons and pred operates on first lon



Use type parameter again since there are no specific type



Checking the type of list produced



(cons X rest) -> need also X



Overview

* Looking at the whole function first
* Then piece by piece we are inferring each type
* We can use type parameters for any type but it must be consistent