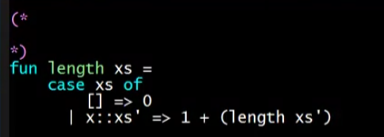
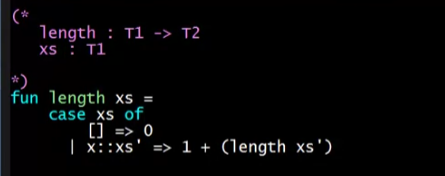


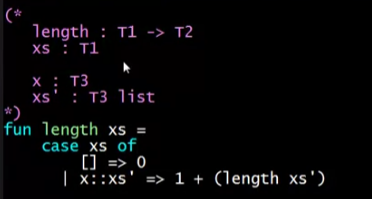
First example: **length** function

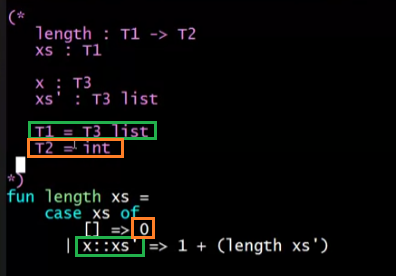


Function definition

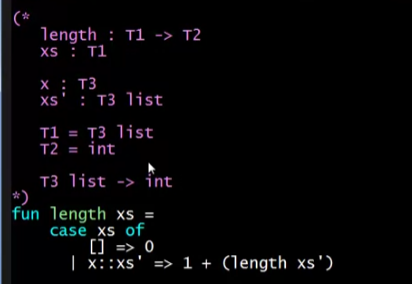


Function body:

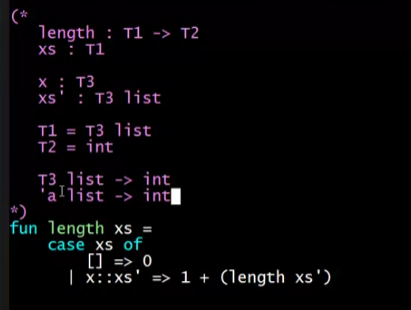




Ending up with

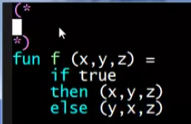


* We cannot infer any more constraints of T3
* We then replace all the instances of this kind of T’s CONSISTENTLY:

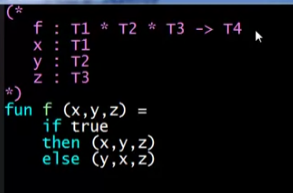


* + If there are T4, it will be ‘b

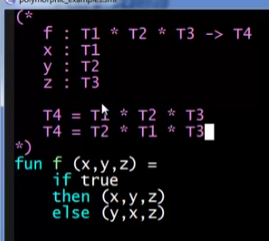
Another example: function f



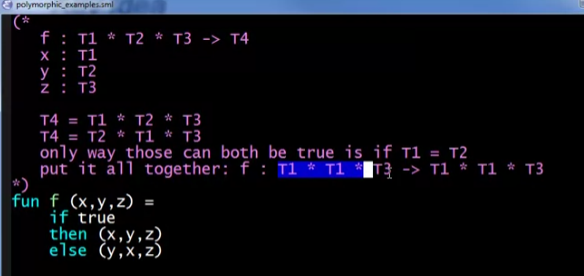
Function definition



Return type (T4)

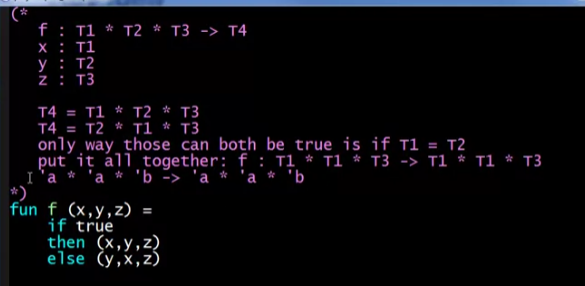


Then for those to type check, x and y must have the same types

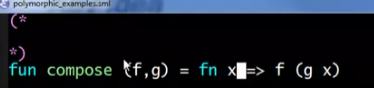


There are no more constraints that can be inferred anymore, so we proceed on consistently replacing type variables

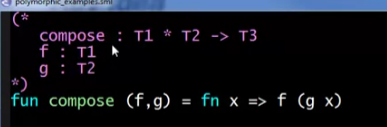
* Every T1 must be ‘a
* Every T3 must be ‘b



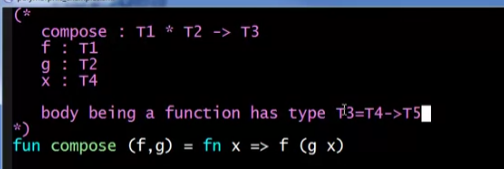
3rd example: function **compose**



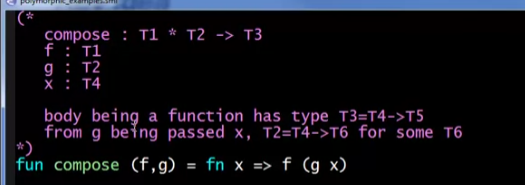
Function definition

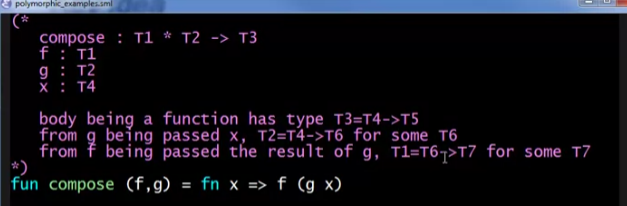


Function body (anonymous function)

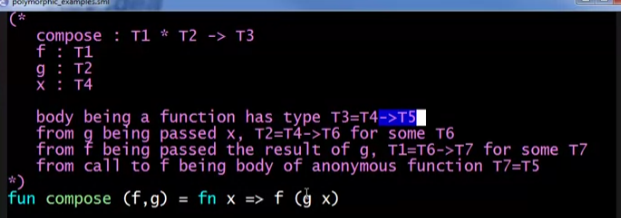


Body of the anonymous function

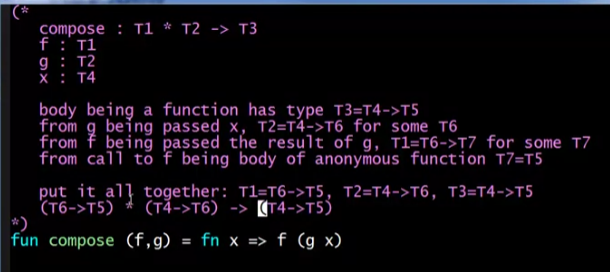




Simplifying the return type of the body of the anonymous function (T7 must be equal to T5), the return type of the body of the whole function



No more constraint, putting it all together:



Replacing with type variable consistently:

* T6 = ‘a
* T5 = ‘b
* T4 = ‘c

