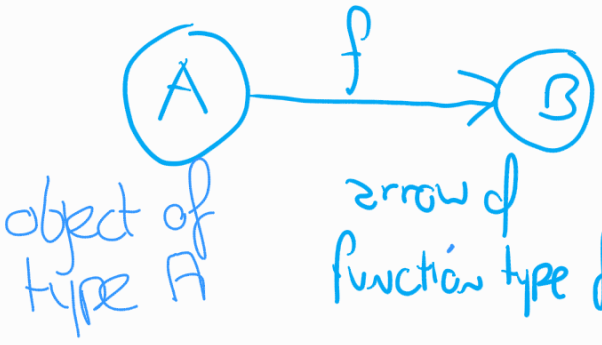
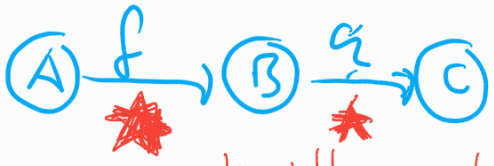


$f :: A \to B$

f : function type; A, B : object types



$f : A \to B$
$g : B \to C$
happy flow : function application

non-happy flow breaks function application : impure / side effect

consider the exception a (side) effect of applying function $f(A) = B$
We want to make the flow 'pure' again.
Solution : encapsulate the side effect ⇒ new functions f' and g' :

$f' :: A \to M[B]$ M encapsulates B
$g' \quad B \to M[C]$ M " C

3 problems: ① impedance mismatch
② $A \to M[B]$ not symmetric with $M[B] \to M[C]$ ⇒ not composable / chainable
③ how to get result C ?

solution to ① start on $M[A]$

solution to (2) define a function flatmap on M
that accept f' (or g') and return M[B]
(or M[C])

flatmap	from M[A]	Given A ⇒ M[B]	To M[B]