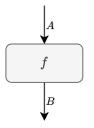
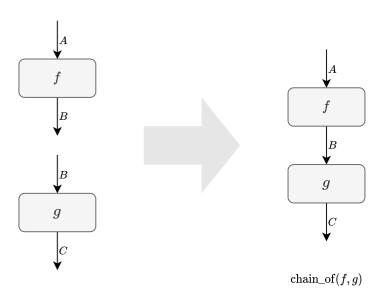
## **Transformation**



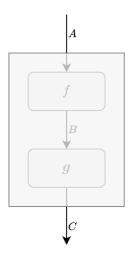
A transformation f maps any input of type A to the output of type B.

# Composition



Transformations with compatible input and output can be composed.

## **Composition is a Transformation**



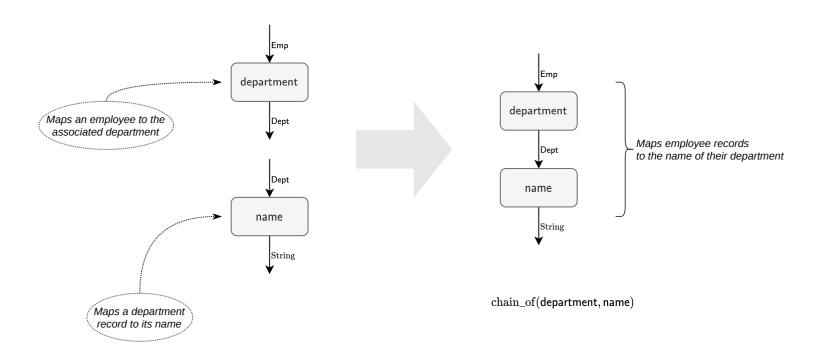
Crucially, composition of transformations is again a transformation.

## **Composition Combinator**

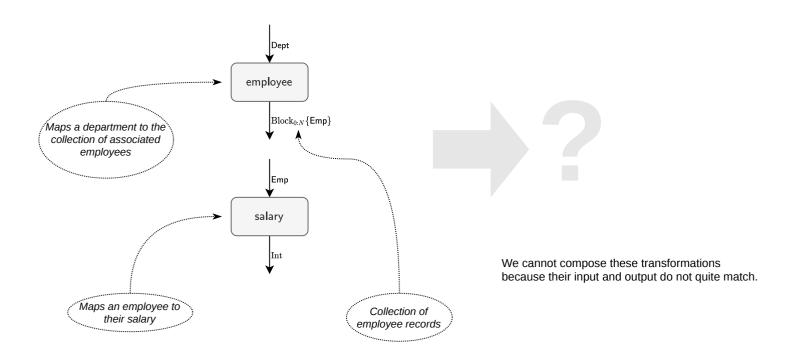


$$\label{eq:composition} \begin{split} & \text{Composition chain\_of}\big([]],[]]\big) \\ & \text{is a transformation combinator} \\ & \text{with two arguments.} \end{split}$$

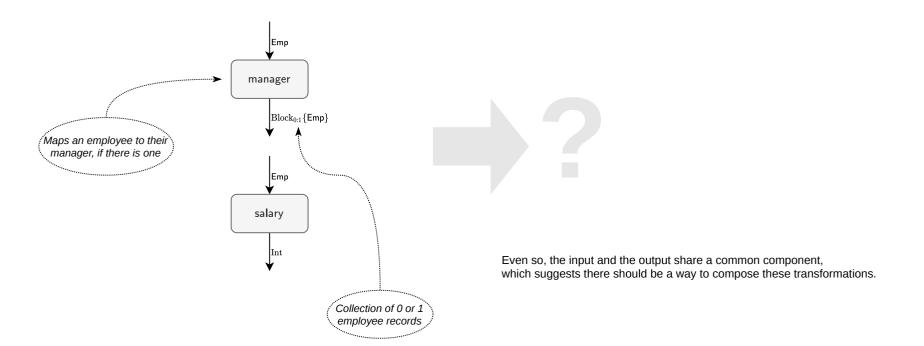
## **Example: Composition**



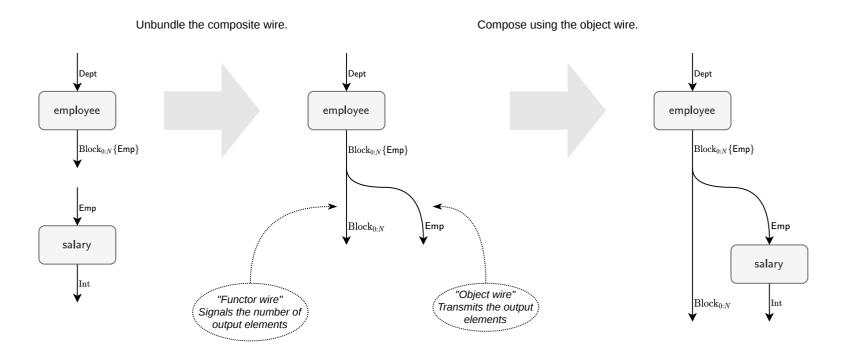
## **Counter-example: Plural Component**



## **Counter-example: Optional Component**



#### Idea: Unbundle the Wire



Attaching a transformation to the object wire indicates that the transformation is applied to all element of the collection.

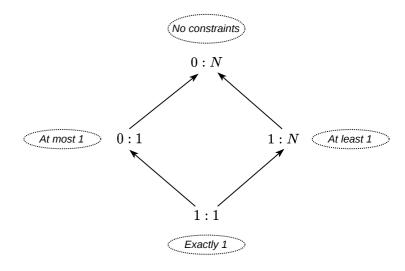
## **Block Type**

A block is a collection of homogeneous elements.

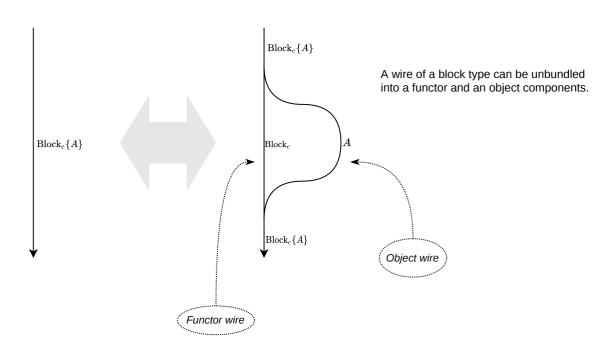
Type of elements

Block $_e\{A\}$ Cardinality of the block

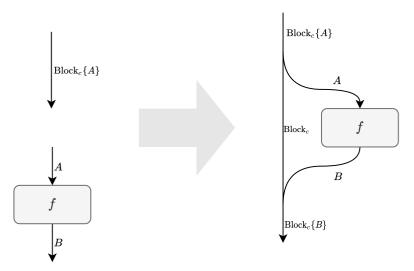
Cardinality is a constraint on the number of elements in a block.



## **Unbundling**



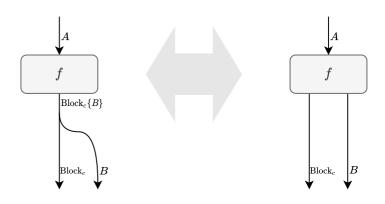
## **Object Transformation**

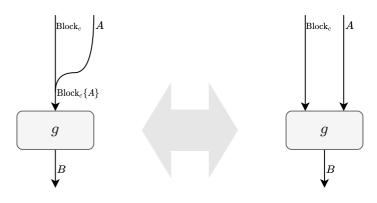


Any compatible transformation can be applied to the object wire, which indicates that the transformation is applied to every element of the block.

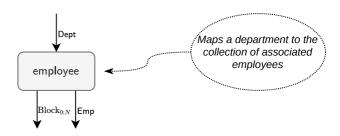
 $with\_elements(f)$ 

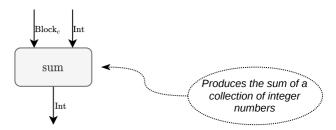
## **Multiwired Transformations**



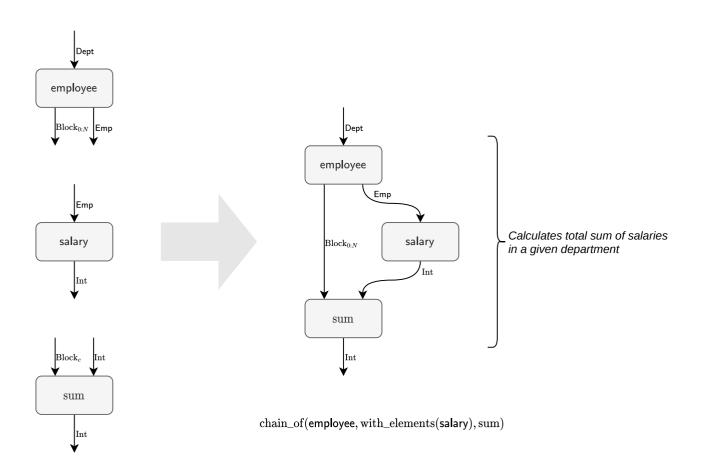


## **Example: Multiwired Transformations**

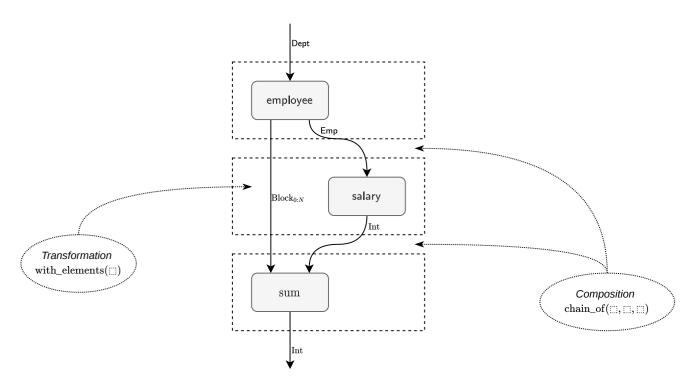




## **Example: Multiwired Composition**



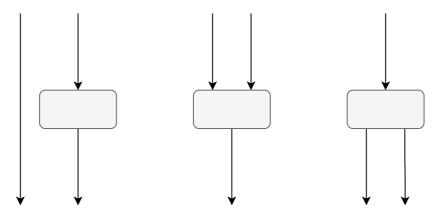
## **Example: Multiwired Composition Details**



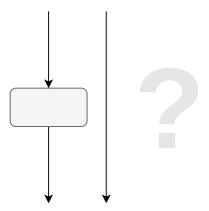
chain\_of(employee, with\_elements(salary), sum)

### **Challenge: Functor Transformation**

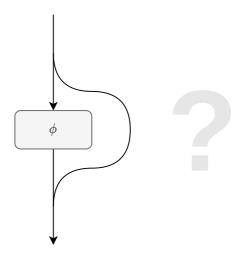
So far we have seen different combinations of transformations and wires: transformations that are applied to the object wire, as well as transformations that consume or produce both functor and object wires.



What we miss is a notion of a transformation applied to the functor wire.

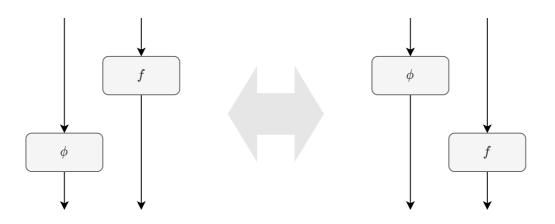


#### **Functor Transformation**



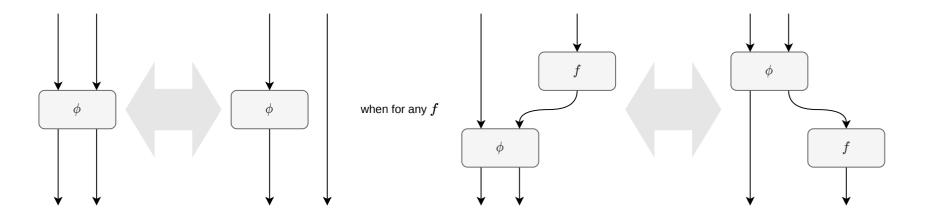
How to define a functor transformation?

Since a functor transformation does not act on the object wire, it cannot interfere with any transformations applied to the object wire.



This property can be taken as a definition of a functor transformation.

#### **Definition: Functor Transformation**



Such transformations are also called *natural*.

Branch of mathematics that studies natural transformations is called the *category theory*.