DATABASES, CATEGORIES AND FUNCTORS

INTRODUCTION TO CATEGORY THEORY

LET'S LOOK AT A DATABASE

Employee	WorksIn	Manager
1	101	2
2	101	3
3	102	3

Department	Secretary
101	2
102	3

LET'S LOOK AT A DATABASE

Employee	WorksIn	Manager
1	101	2
2	101	3
3	102	3

Department	Secretary
101	2
102	3

Schemas

EMPLOYEES

Employee: EmployeeID
WorksIn: DepartmentID
Manager: EmployeeID

DEPARTMENTS

Department: DepartmentID
Secretary: EmployeeID

DATABASE SCHEMA AS A GRAPH

EMPLOYEES

Employee: EmployeeID
WorksIn: DepartmentID
Manager: EmployeeID

DEPARTMENTS

Department: DepartmentID
Secretary: EmployeeID

DATABASE SCHEMA AS A GRAPH

EMPLOYEES

Employee: EmployeeID
WorksIn: DepartmentID
Manager: EmployeeID

DEPARTMENTS

Department: DepartmentID
Secretary: EmployeeID

Each ID column becomes a node (vertex)

Employee

Department

DATABASE SCHEMA AS A GRAPH

EMPLOYEES

Employee: EmployeeID
WorksIn: DepartmentID
Manager: EmployeeID

DEPARTMENTS

Department: DepartmentID
Secretary: EmployeeID

Each non-ID column becomes an edge

DATABASE SCHEMA AS A GRAPH: IMPOSING EXTRA RULES

EMPLOYEES

Employee: EmployeeID
WorksIn: DepartmentID
Manager: EmployeeID

DEPARTMENTS

Department: DepartmentID
Secretary: EmployeeID

Each non-ID column becomes an edge

Department.Secretary.Department = Department

Employee.Manager.WorksIn = Employee.WorksIn

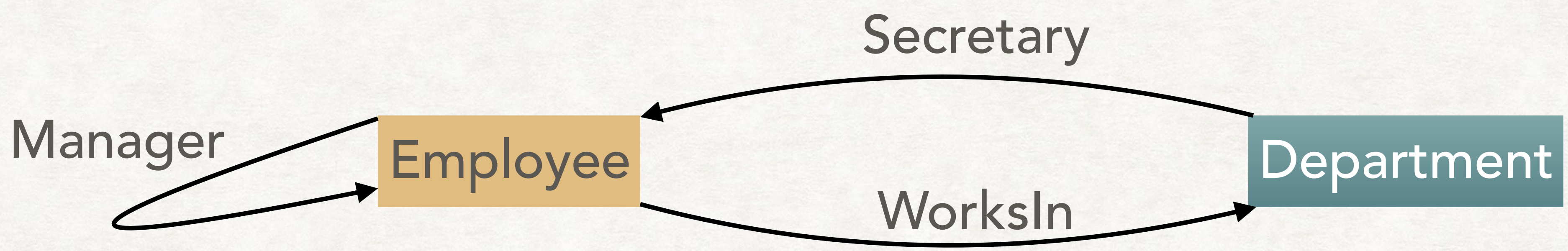
ANOTHER EXAMPLE

Schema 1

Schema 2

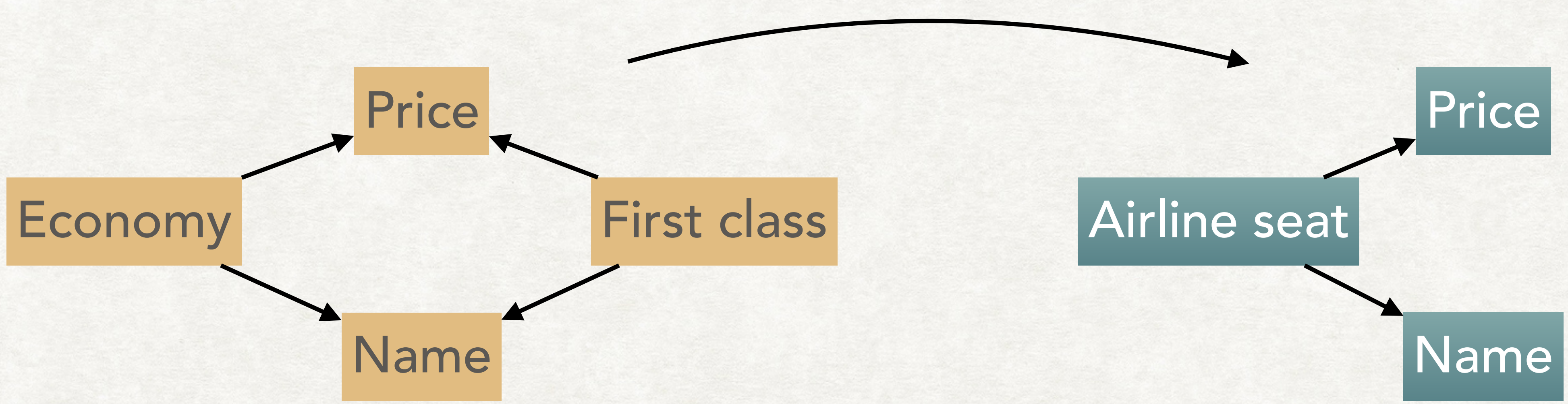
How does one migrate data between two databases?

DATABASE SCHEMA AS A GRAPH: SO WHAT?

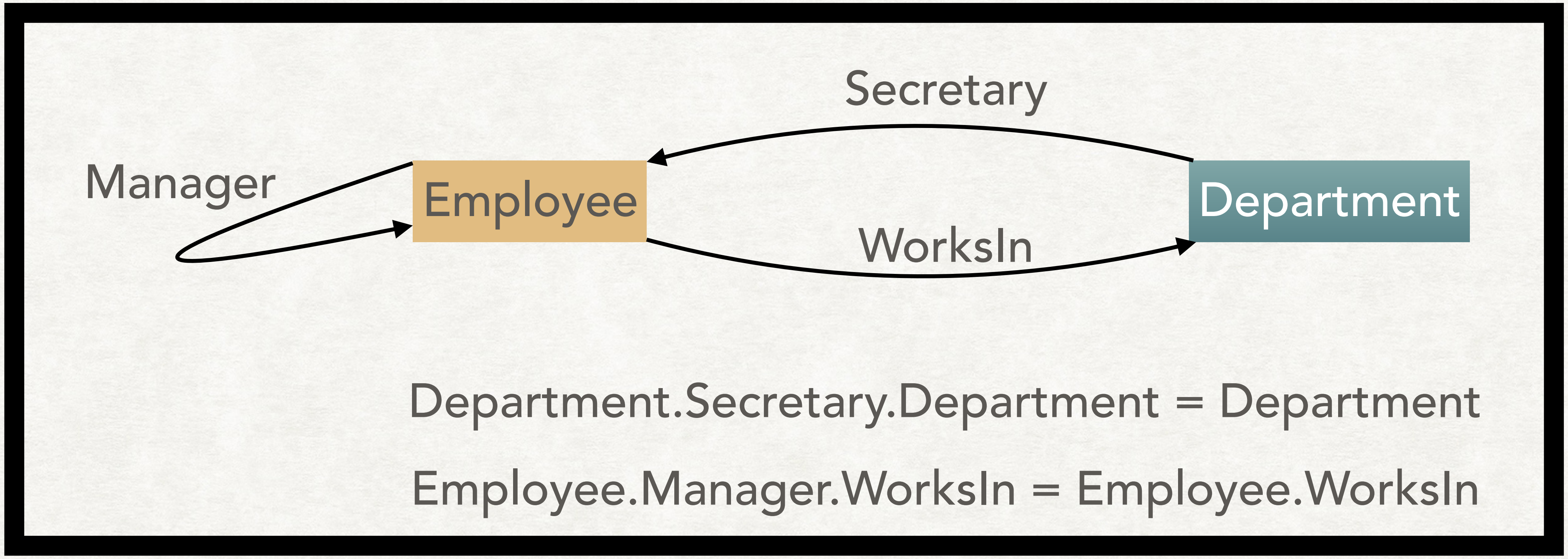


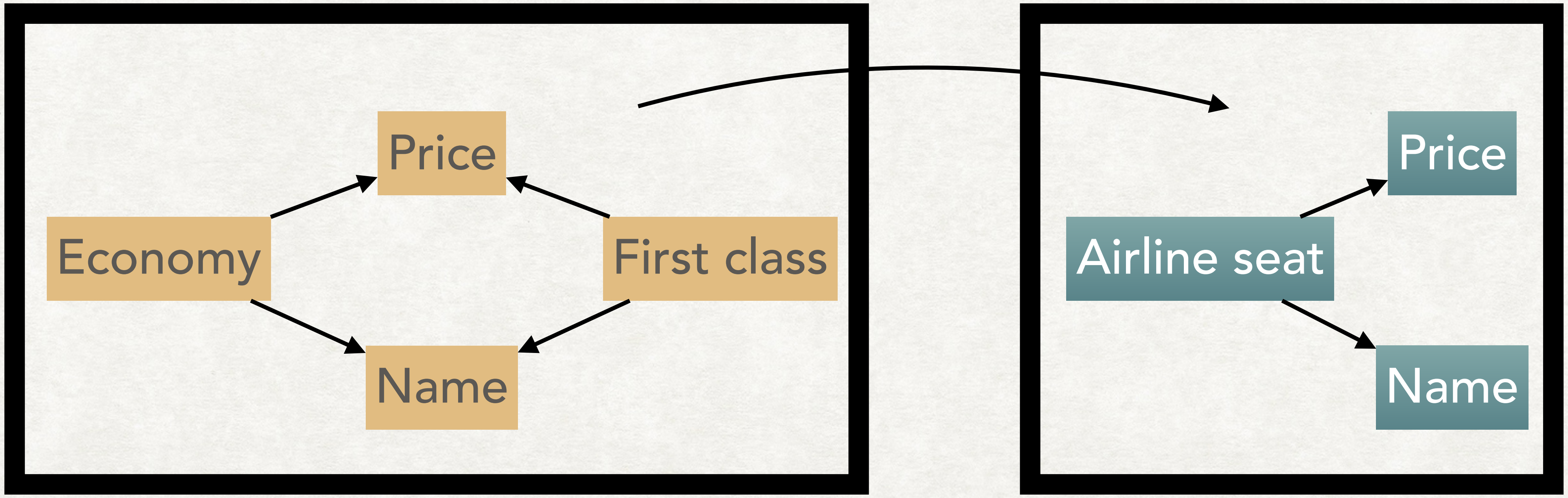
Department.Secretary.Department = Department

Employee.Manager.WorksIn = Employee.WorksIn

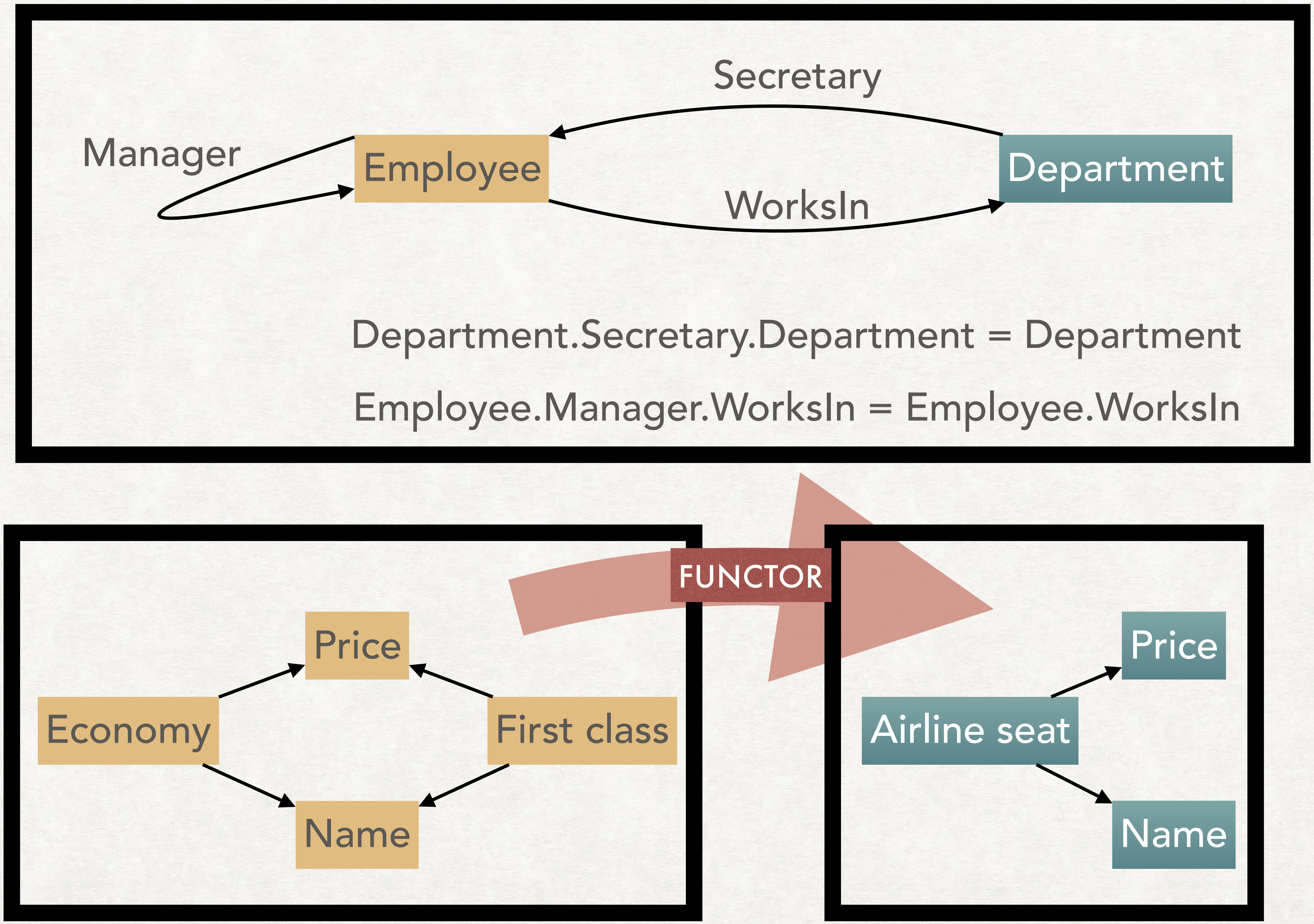


THESE PICTURES ARE CATEGORIES

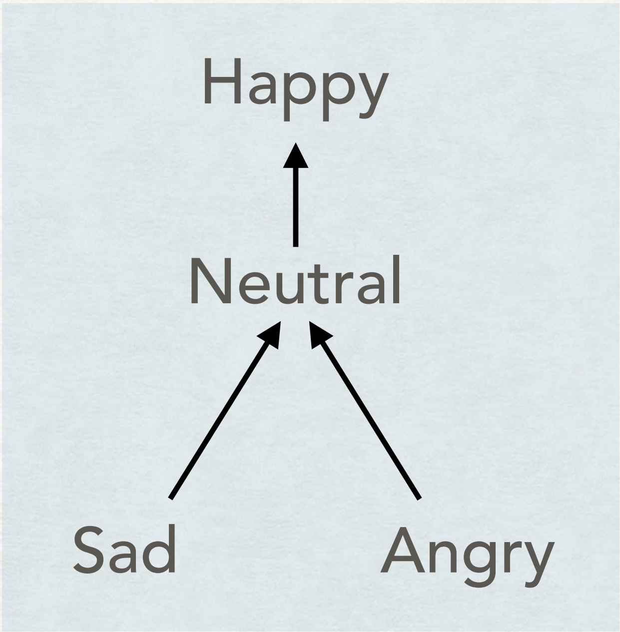




ARROW IS FUNCTOR

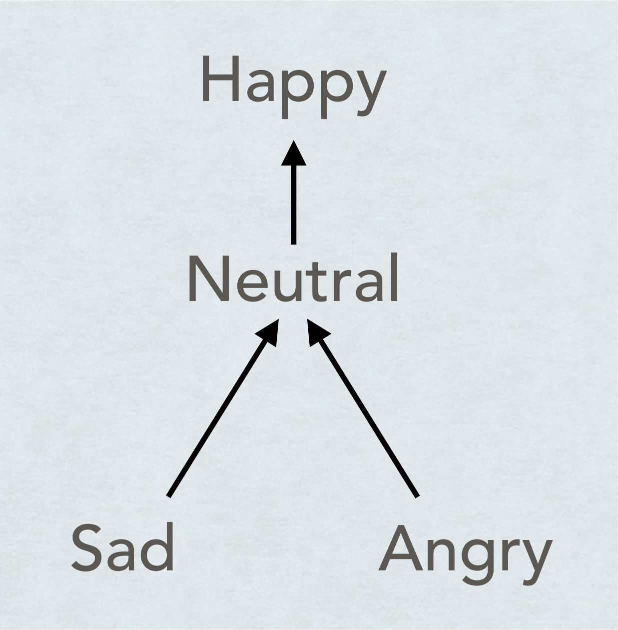


WHAT IS A CATEGORY? LET'S START WITH FAMILIAR EXAMPLES



Partial order

WHAT IS A CATEGORY? LET'S START WITH FAMILIAR EXAMPLES



Partial order

COLLECTION OF OBJECTS

Happy, Neutral, Sad, Angry

EVERY PAIR OF OBJECTS HAS A "RELATIONSHIP"

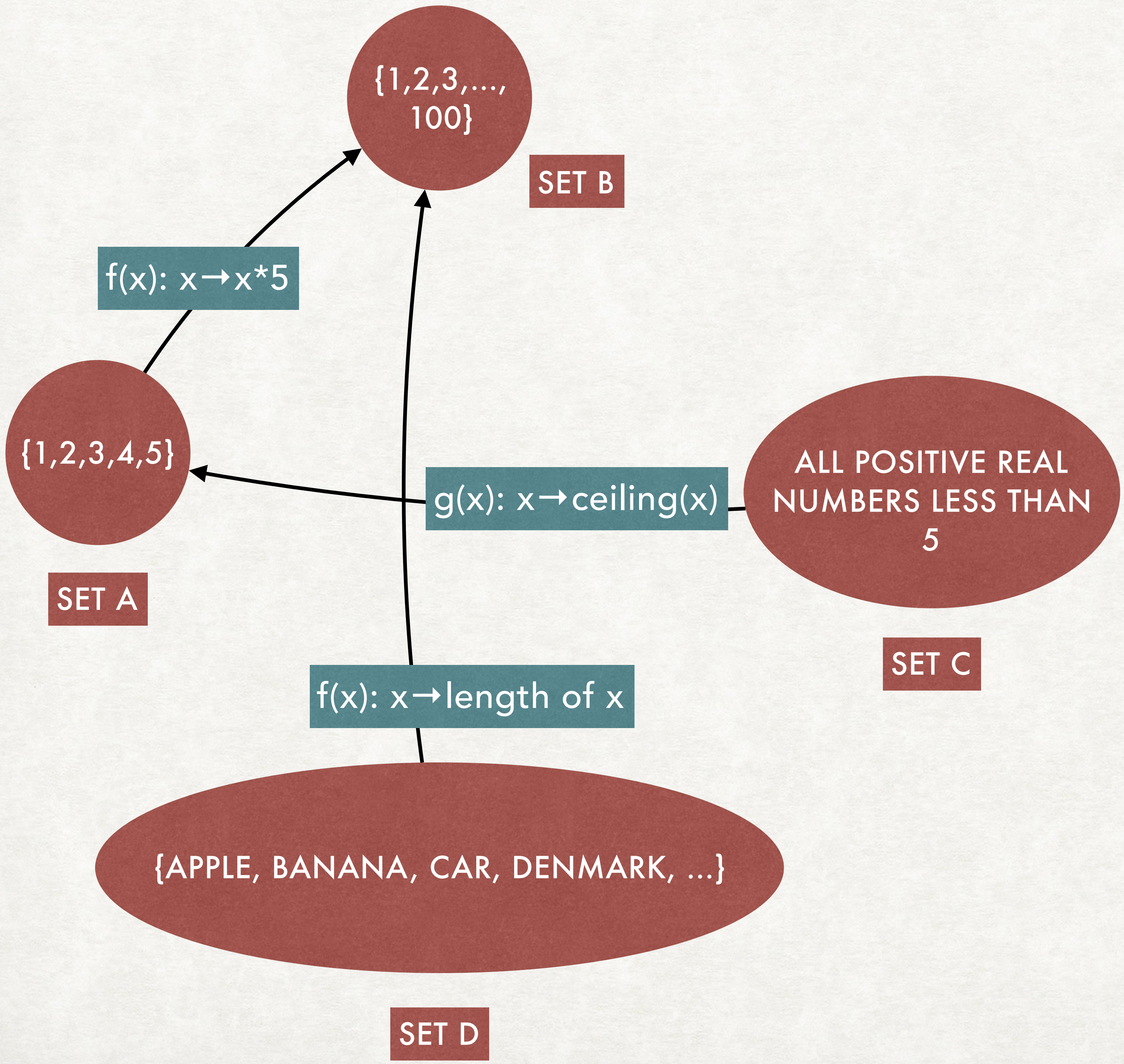
Sad, Neutral: ≤

Neutral, Happy: ≤

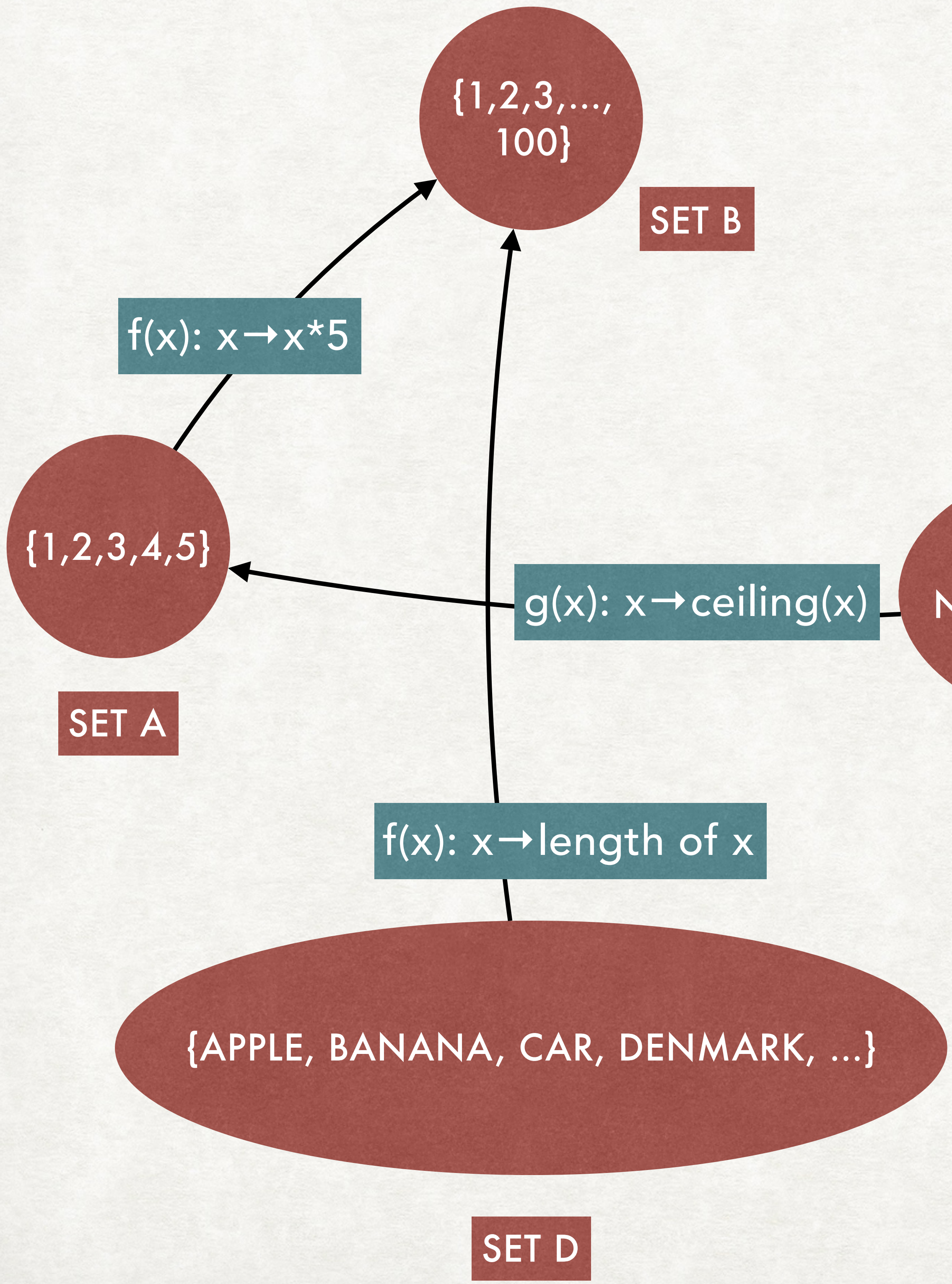
Sad, Angry: Nothing

NOTHING IS ALSO A "RELATIONSHIP"

WHAT IS A CATEGORY? LET'S START WITH FAMILIAR EXAMPLES



WHAT IS A CATEGORY? LET'S START WITH FAMILIAR EXAMPLES



COLLECTION OF OBJECTS

Set A, Set B, Set C, Set D

EVERY PAIR OF OBJECTS HAS A "RELATIONSHIP"

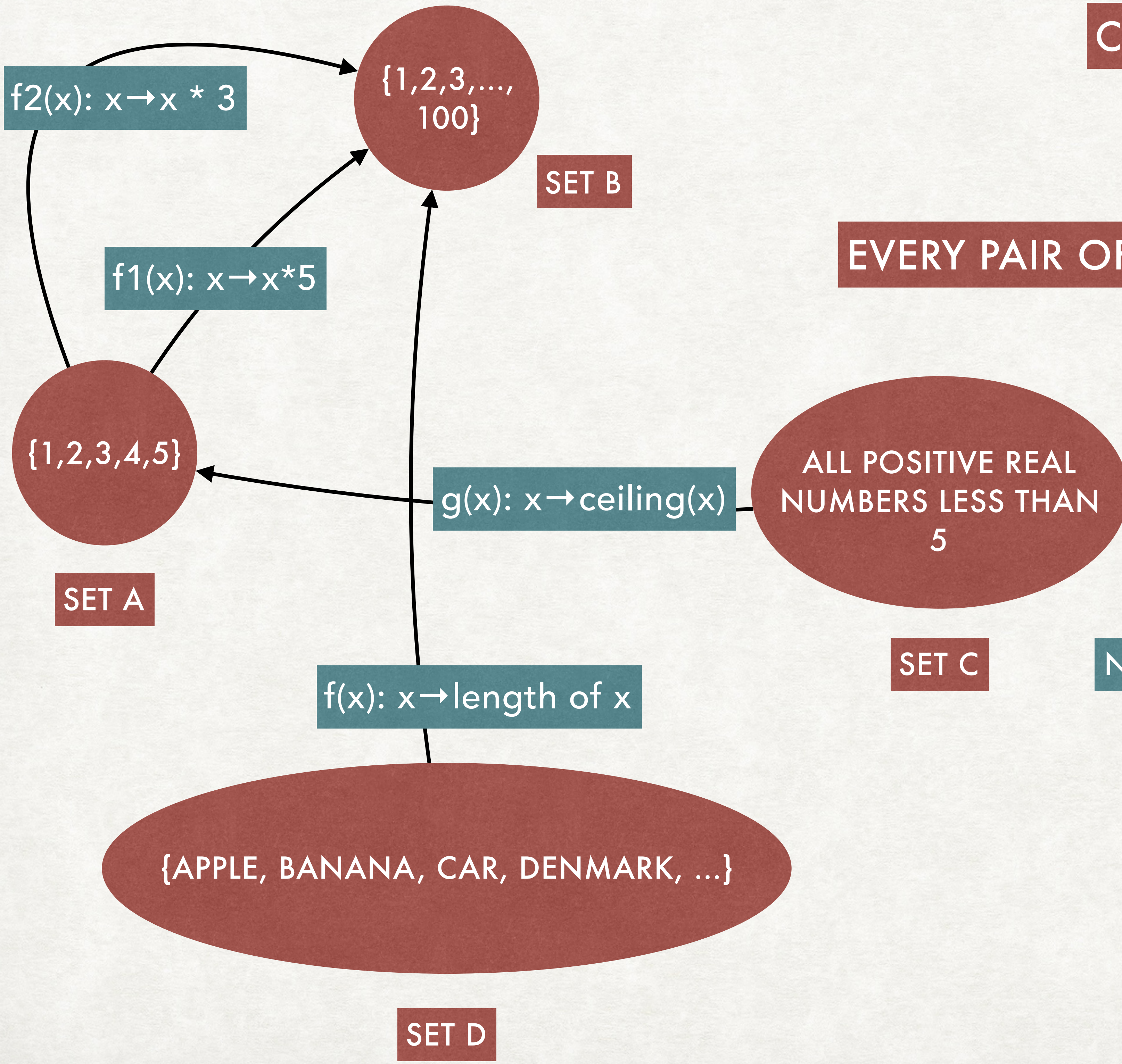
Set A, Set B: f

Set C, Set A: g

Set B, Set C: Nothing

NOTHING IS ALSO A "RELATIONSHIP"

WHAT IS A CATEGORY? LET'S START WITH FAMILIAR EXAMPLES



COLLECTION OF OBJECTS

Set A, Set B, Set C, Set D

EVERY PAIR OF OBJECTS HAS A "RELATIONSHIP"

Set A, Set B: {f1, f2}

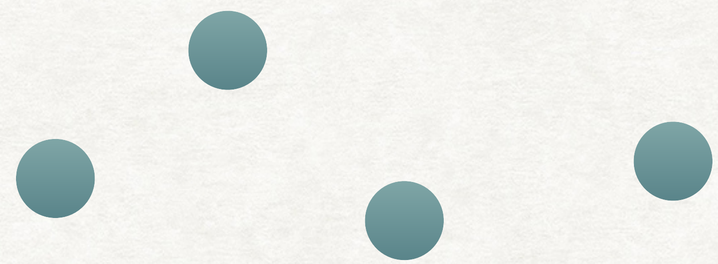
Set C, Set A: {g}

Set B, Set C: ∅

NOTHING IS ALSO A "RELATIONSHIP"

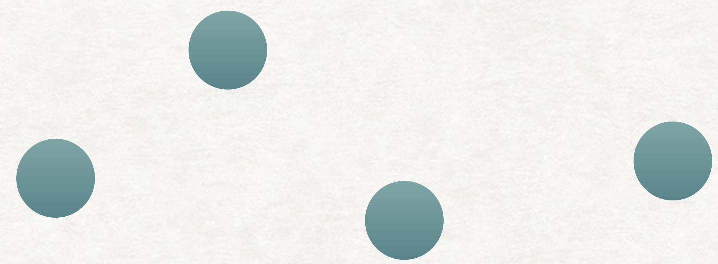
WHAT IS A CATEGORY? DEFINITION

COLLECTION OF OBJECTS

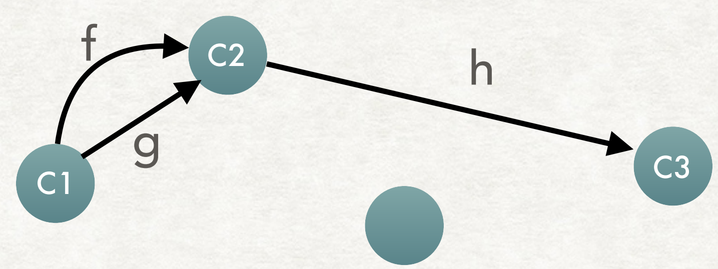


WHAT IS A CATEGORY? DEFINITION

COLLECTION OF OBJECTS



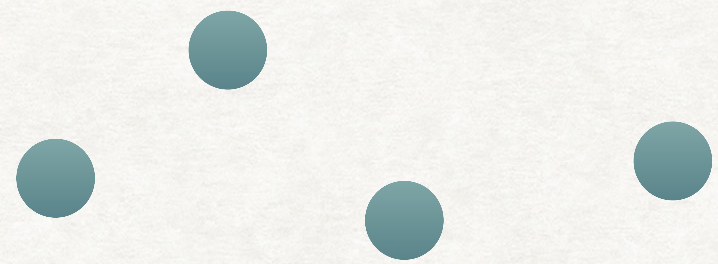
EVERY PAIR OF OBJECTS A AND B HAS A “RELATIONSHIP”:
MORPHISM BETWEEN A AND B



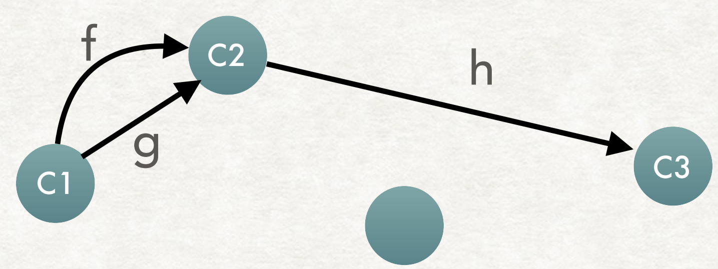
Set of morphisms
between C1 and C2
= {f1, f2}

WHAT IS A CATEGORY? DEFINITION

COLLECTION OF OBJECTS

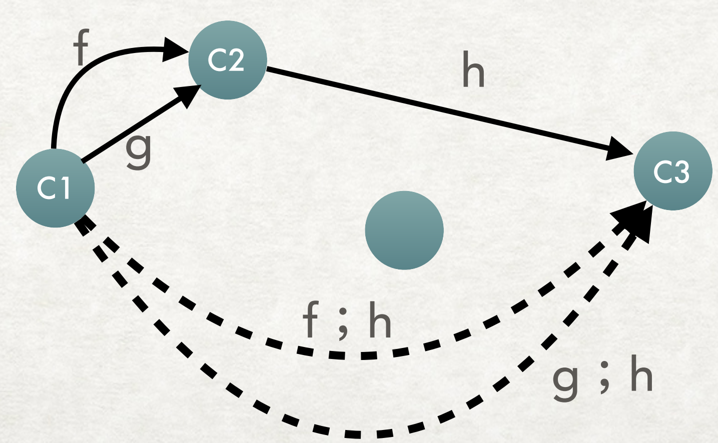


EVERY PAIR OF OBJECTS A AND B HAS A "RELATIONSHIP": MORPHISM BETWEEN A AND B



Set of morphisms between C1 and C2 = {f, g}

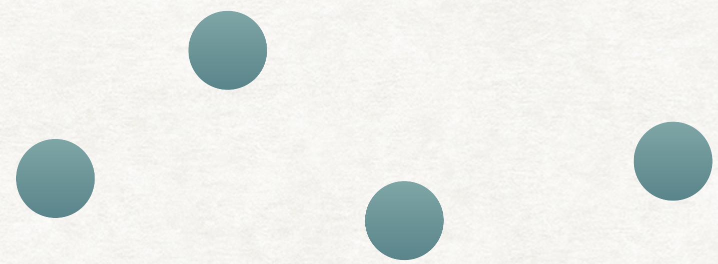
TWO CONSECUTIVE MORPHISMS CAN BE COMPOSED TOGETHER



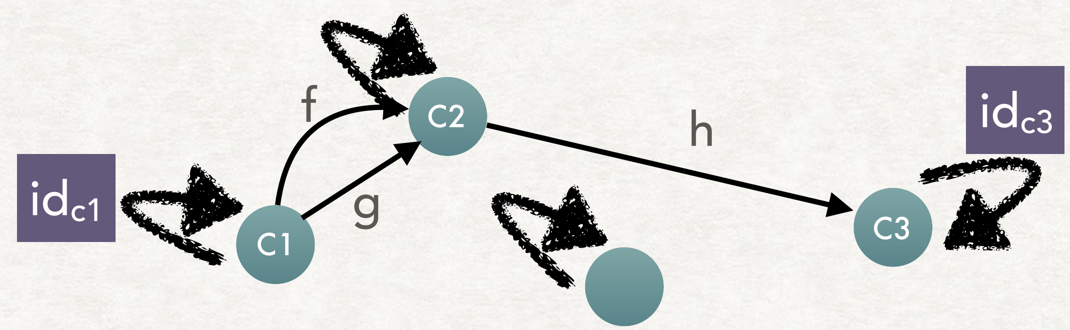
Composition of morphisms f and h = A morphism between C1 and C3

WHAT IS A CATEGORY? DEFINITION

COLLECTION OF OBJECTS



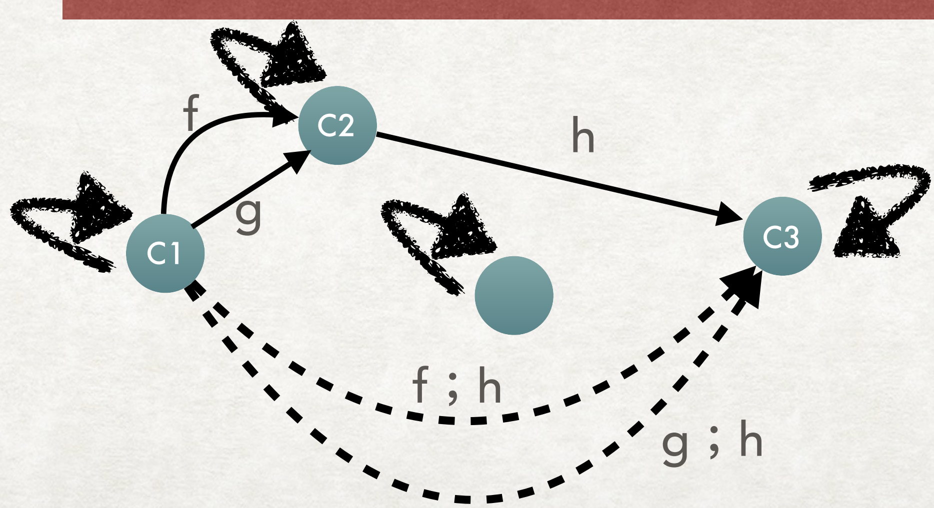
THERE IS A SPECIAL MORPHISM BETWEEN AN ELEMENT AND ITSELF:
MORPHISM (C1, C1) = IDENTITY MORALISM OF C1



id_{c1} ; f = f

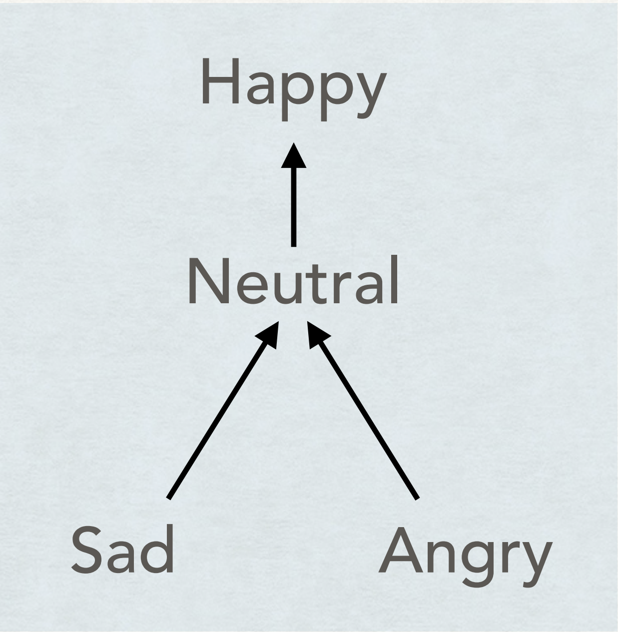
f ; id_{c2} = f

MORPHISMS ARE ASSOCIATIVE



(f1 ; f2) ; f3 = f1 ; (f2 ; f3)

WHAT IS A CATEGORY? LET'S START WITH FAMILIAR EXAMPLES



Partial order

COLLECTION OF OBJECTS

Happy, Neutral, Sad, Angry

EVERY PAIR OF OBJECTS HAS A MORPHISM

Sad, Neutral: {≤}

Neutral, Happy: {≤}

Sad, Angry: ∅

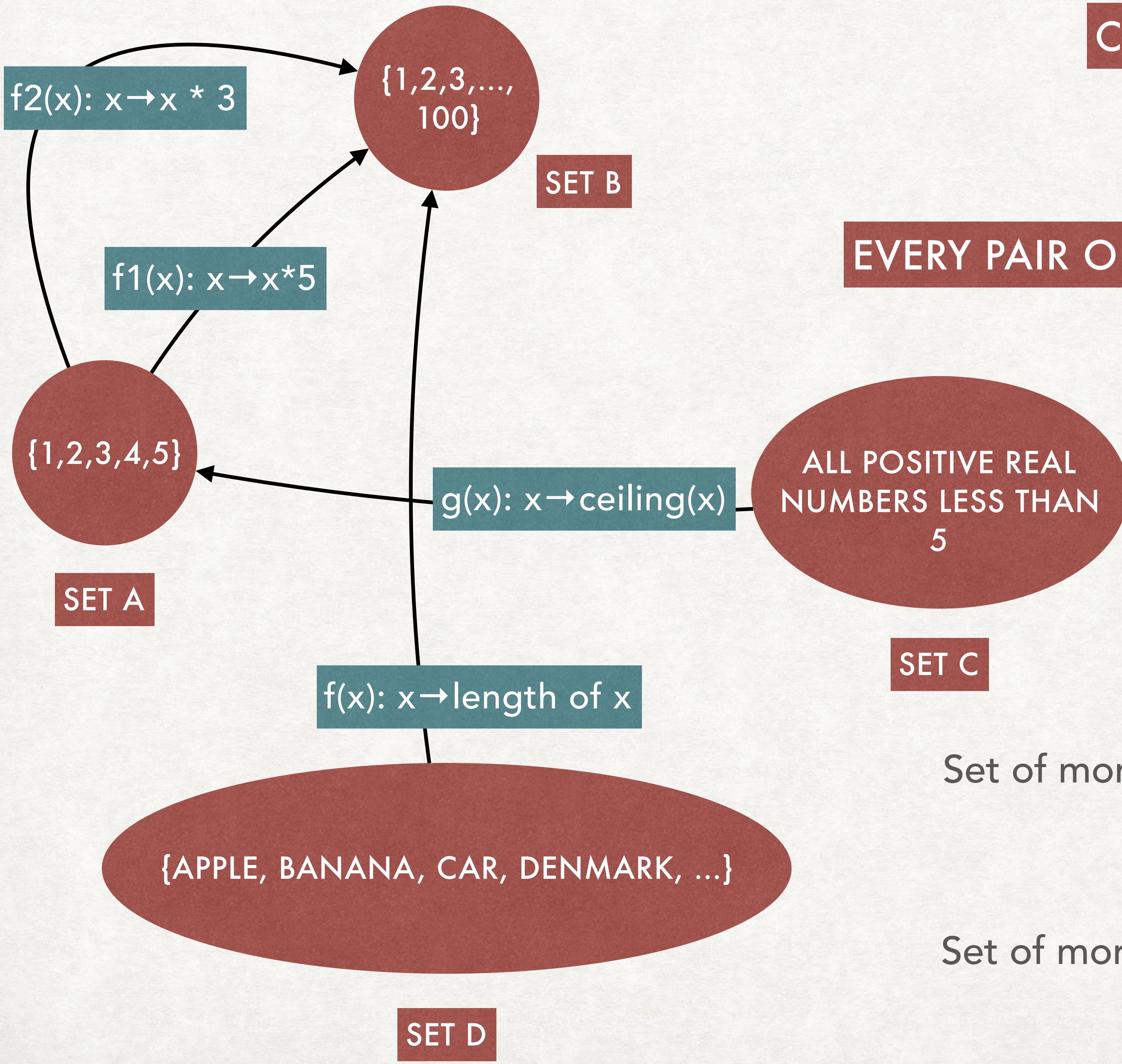
PARTIAL ORDERS HAVE ONLY ONE KIND OF MORPHISM: ≤

Id_{sad} = Morphism (Sad, Sad) = {≤}

Morphism (Sad, Neutral) ; Morphism (Neutral, Happy) =
Morphism (Sad, Happy)

Sad ≤ Neutral and Neutral ≤ Happy
implies
Sad ≤ Happy

WHAT IS A CATEGORY? LET'S START WITH FAMILIAR EXAMPLES



COLLECTION OF OBJECTS

Set A, Set B, Set C, Set D

EVERY PAIR OF SETS HAS A SET OF MORPHISMS

Set A, Set B: {f1, f2}

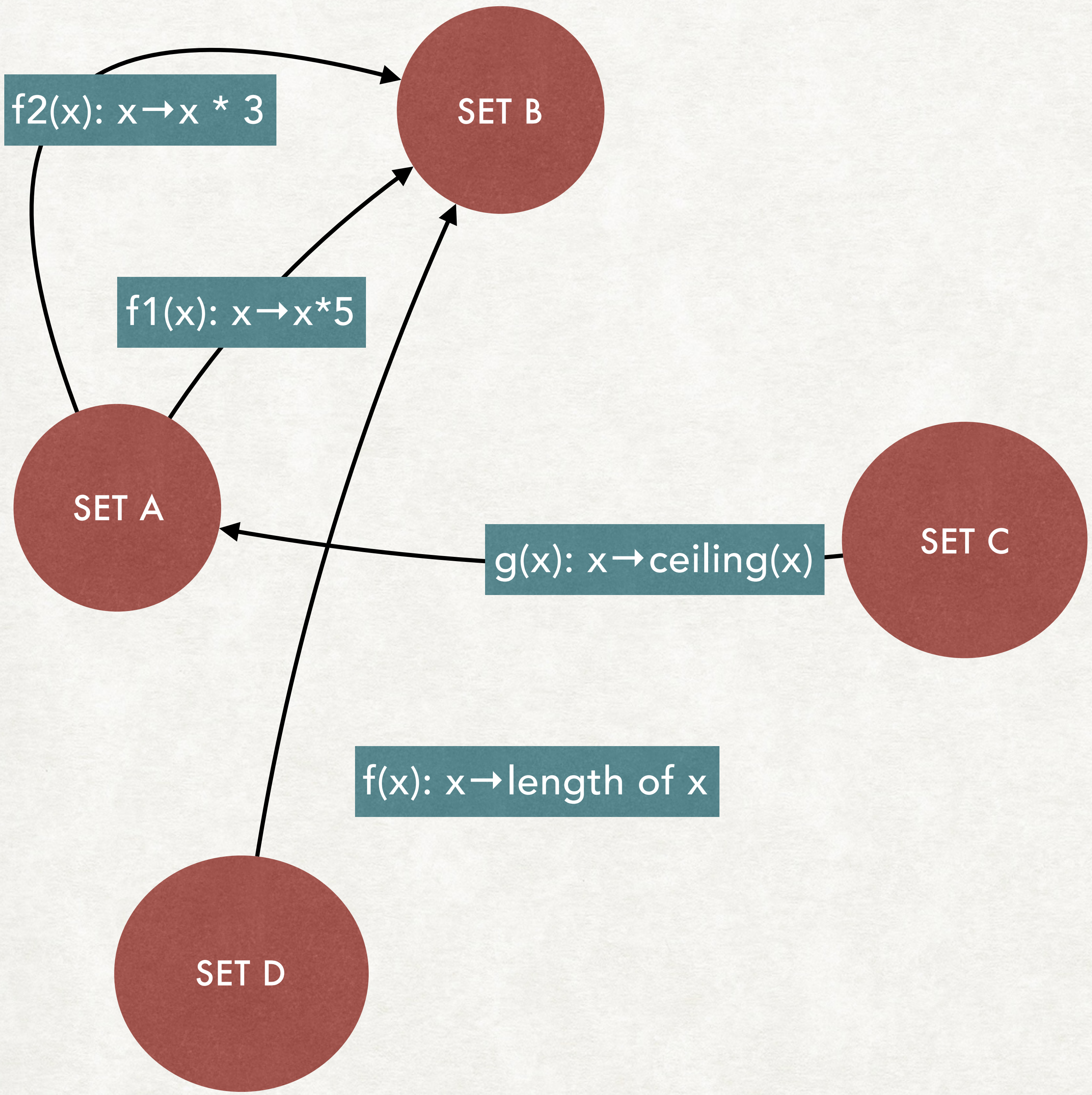
Set C, Set A: {g}

Set B, Set C: ∅

Set of morhpisms between (Set C, Set B) = {g ; f1, g ; f2}

Set of morhpisms between (Set A, Set A) = $id_A = x \to x$

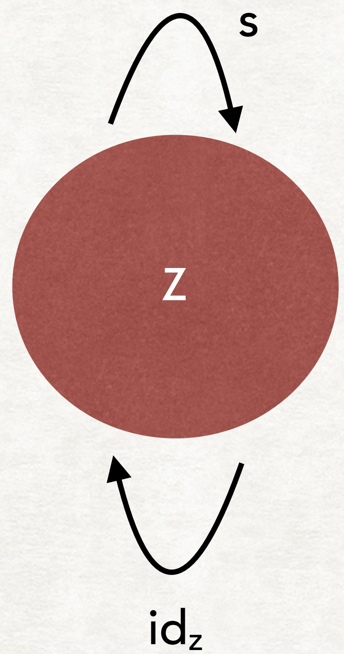
WHAT IS A CATEGORY? LET'S START WITH FAMILIAR EXAMPLES



WHEN YOU THINK OF OBJECTS IN A CATEGORY, YOU DON'T CARE WHAT'S INSIDE EACH OBJECT.

YOU ONLY CARE ABOUT RELATIONSHIPS BETWEEN OBJECTS

CATEGORY WITH A SINGLE OBJECT



How many morphisms does it have?

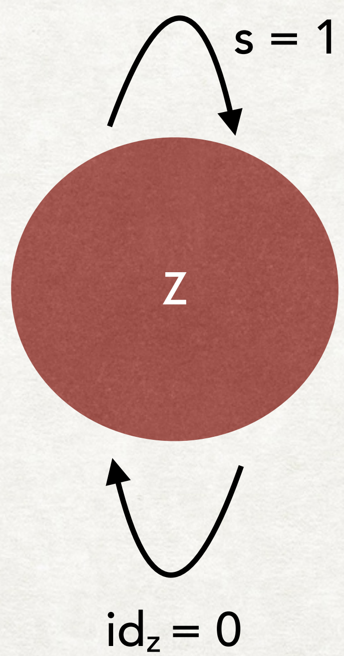
$s = s \,;\, id_z = id_z \,;\, s$

$s \,;\, s$

$s \,;\, s \,;\, s$

$s \,;\, s \,;\, s \,;\, s$

$\circ \;\; \circ \;\; \circ$



Define $a \,;\, b = a + b$

How many morphisms does it have?

$1 = 1 \,;\, 0 = 0 \,;\, 1$

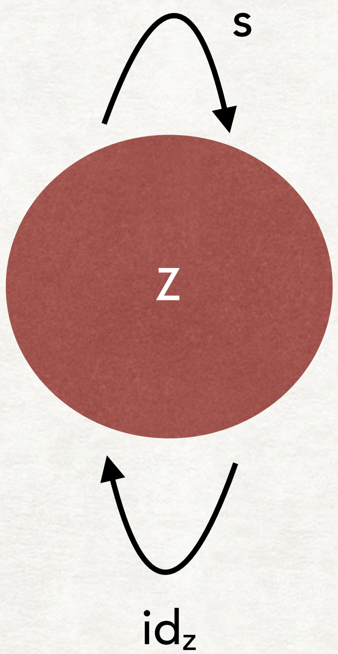
$1 \,;\, 1 = 2$

$1 \,;\, 1 \,;\, 1 = 2 \,;\, 1 = 3$

$1 \,;\, 1 \,;\, 1 \,;\, 1 = 4$

$\circ \;\; \circ \;\; \circ$

CATEGORY WITH A SINGLE OBJECT: WE CAN CONSTRAIN MORPHISMS



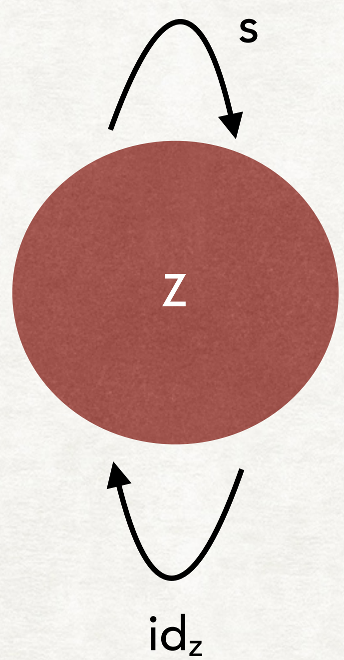
$s ; s = s$

How many morphisms does it have?

$$s = s ; \mathrm{id}_Z = \mathrm{id}_Z ; s$$
$$s ; s = s$$
$$s ; s ; s = s ; s = s$$

Thus it has only two morphisms: id_Z and s

CATEGORY WITH A SINGLE OBJECT: WE CAN CONSTRAIN MORPHISMS

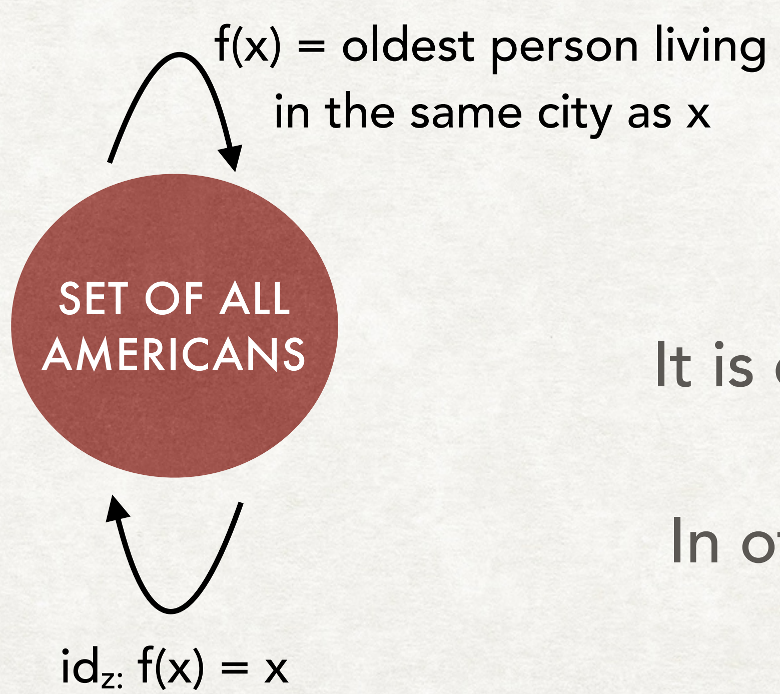


$s ; s = s$

How many morphisms does it have?

$$s = s ; id_z = id_z ; s$$
$$s ; s = s$$
$$s ; s ; s = s ; s = s$$

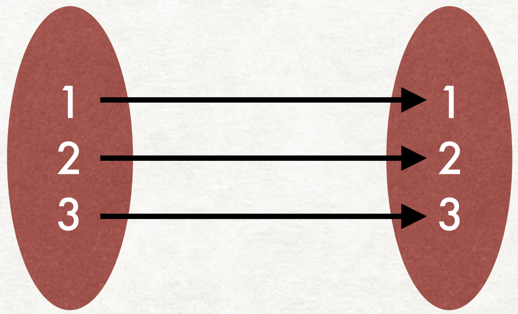
Thus it has only two morphisms: id_z and s



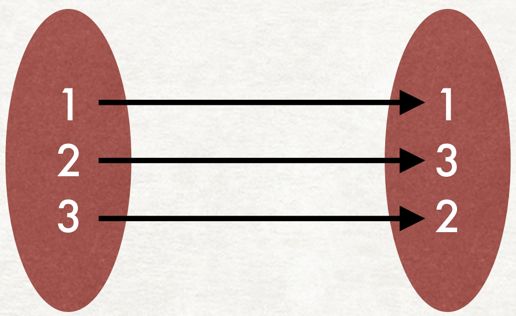
It is easy to see that $f ; f = f$

In other words, $f(f(x)) = f(x)$

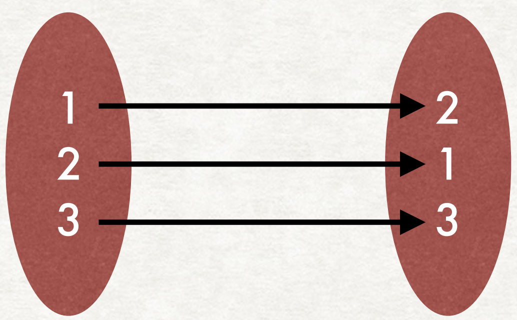
WHAT IS A CATEGORY? PERMUTATION GROUP AS A CATEGORY



f1

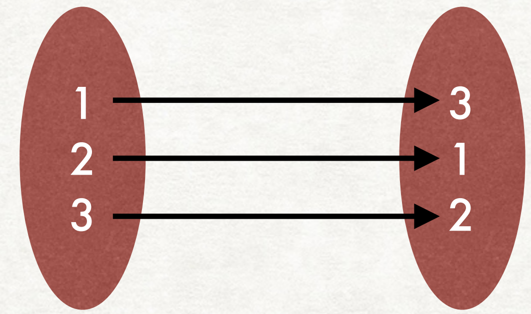


f2

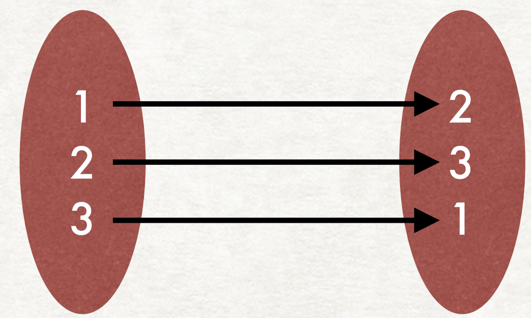


f3

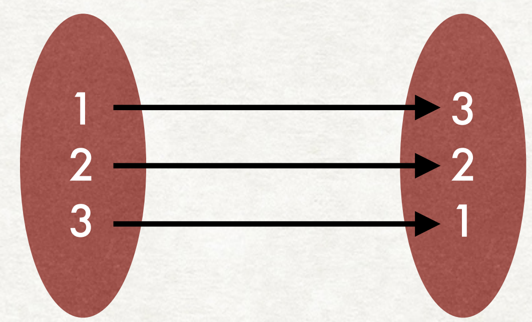
f4



f5

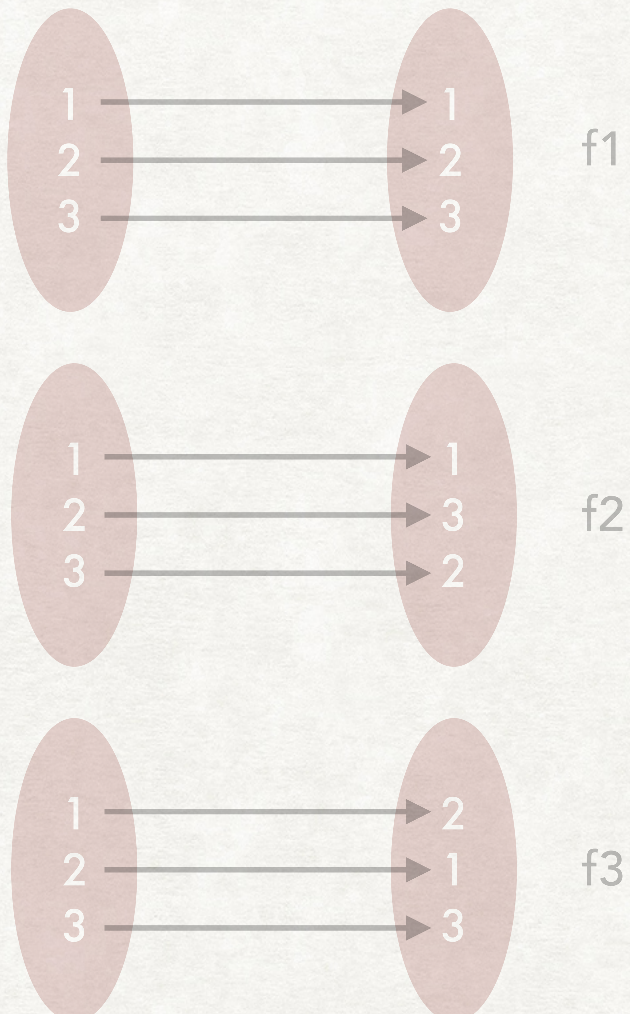


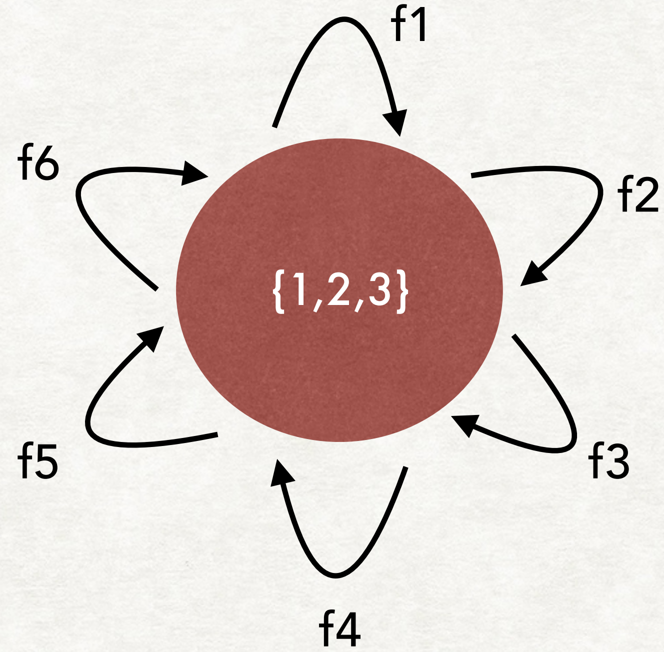
f6

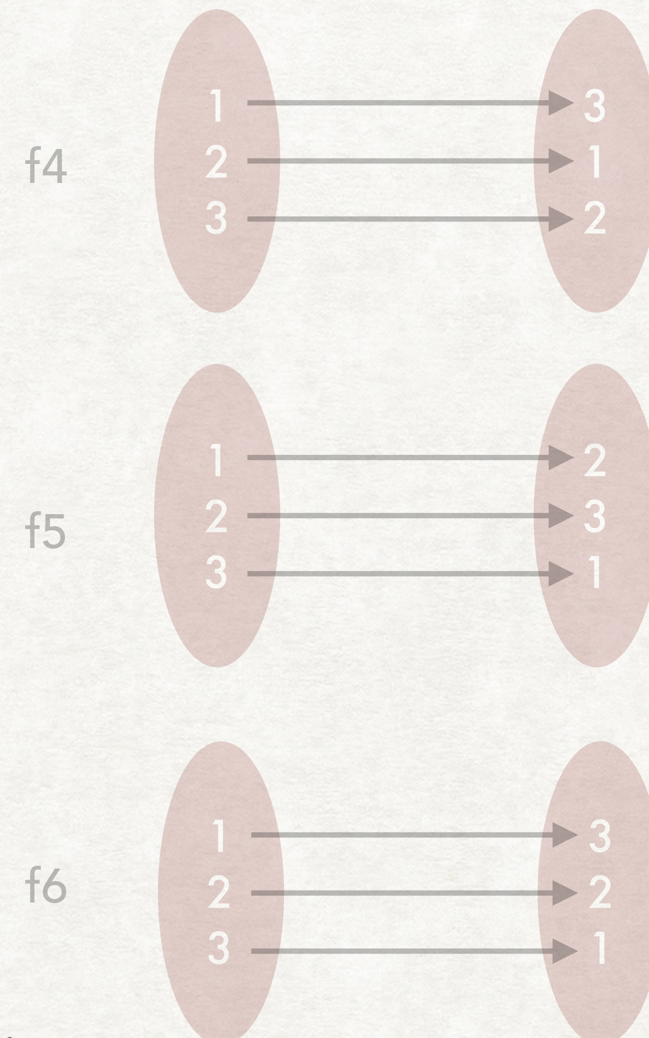


{f1, f2, f3, f4, f5, f6}
is a group of permutations

WHAT IS A CATEGORY? PERMUTATION GROUP AS A CATEGORY







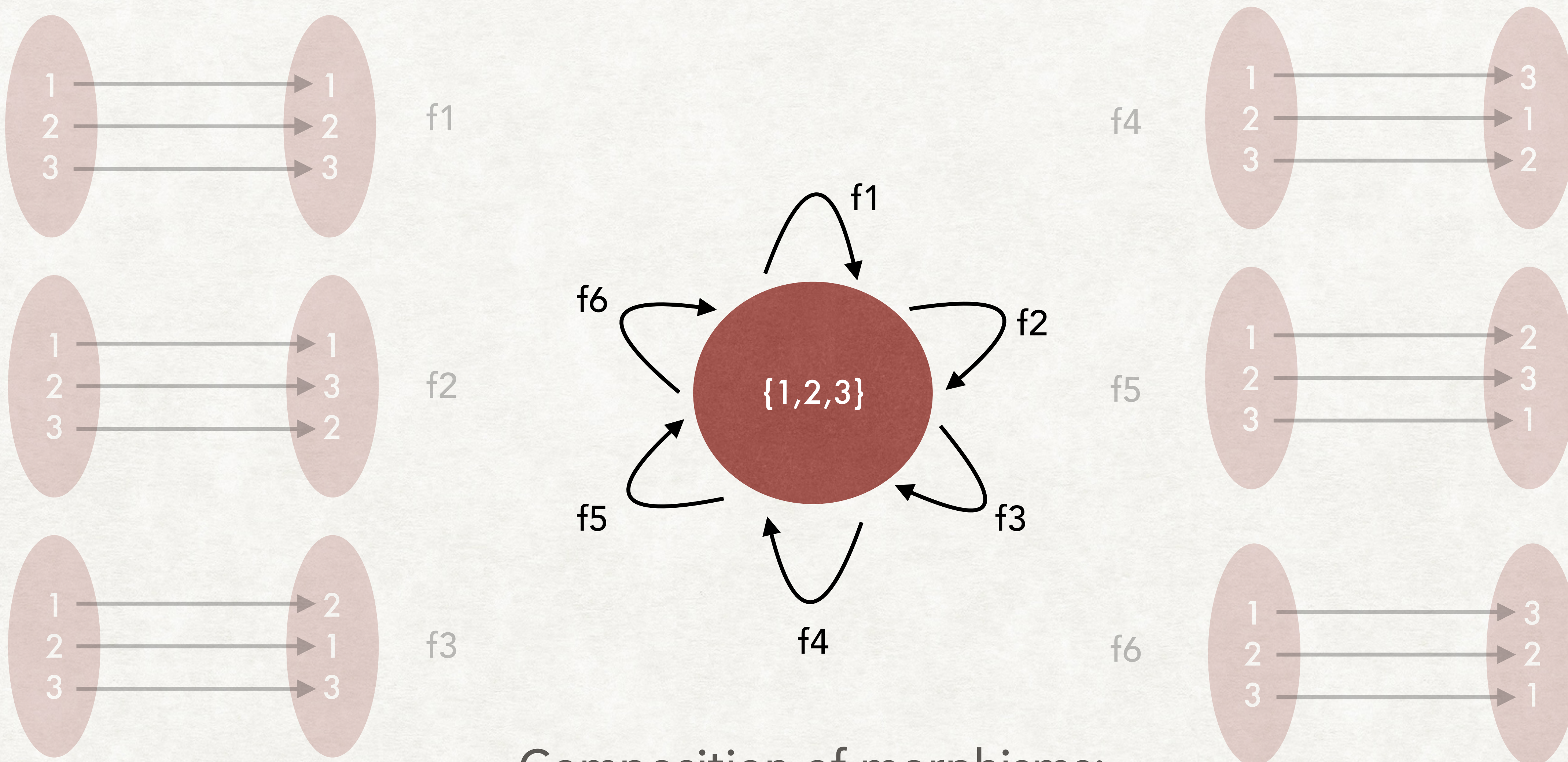
Composition of morphisms:

f5 ; f2 = f6

{f1, f2, f3, f4, f5, f6}

is a group of permutations

WHAT IS A CATEGORY? PERMUTATION GROUP AS A CATEGORY

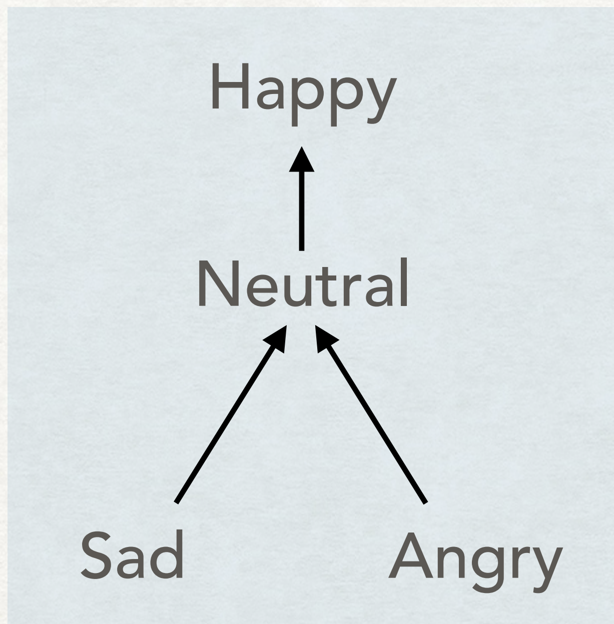


Composition of morphisms:

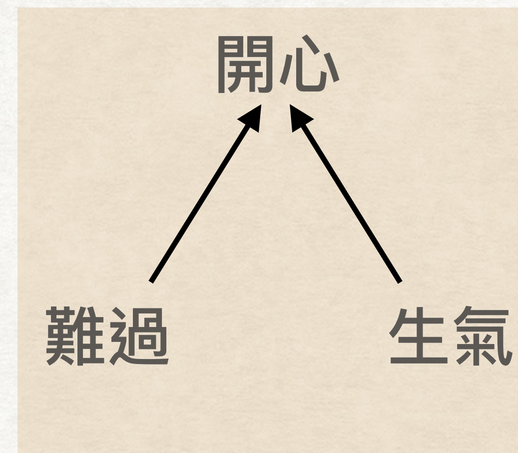
f5 ; f2 = f6

Category with one object and six morphisms

WHAT IS A FUNCTOR? LET'S START WITH A FAMILIAR EXAMPLE

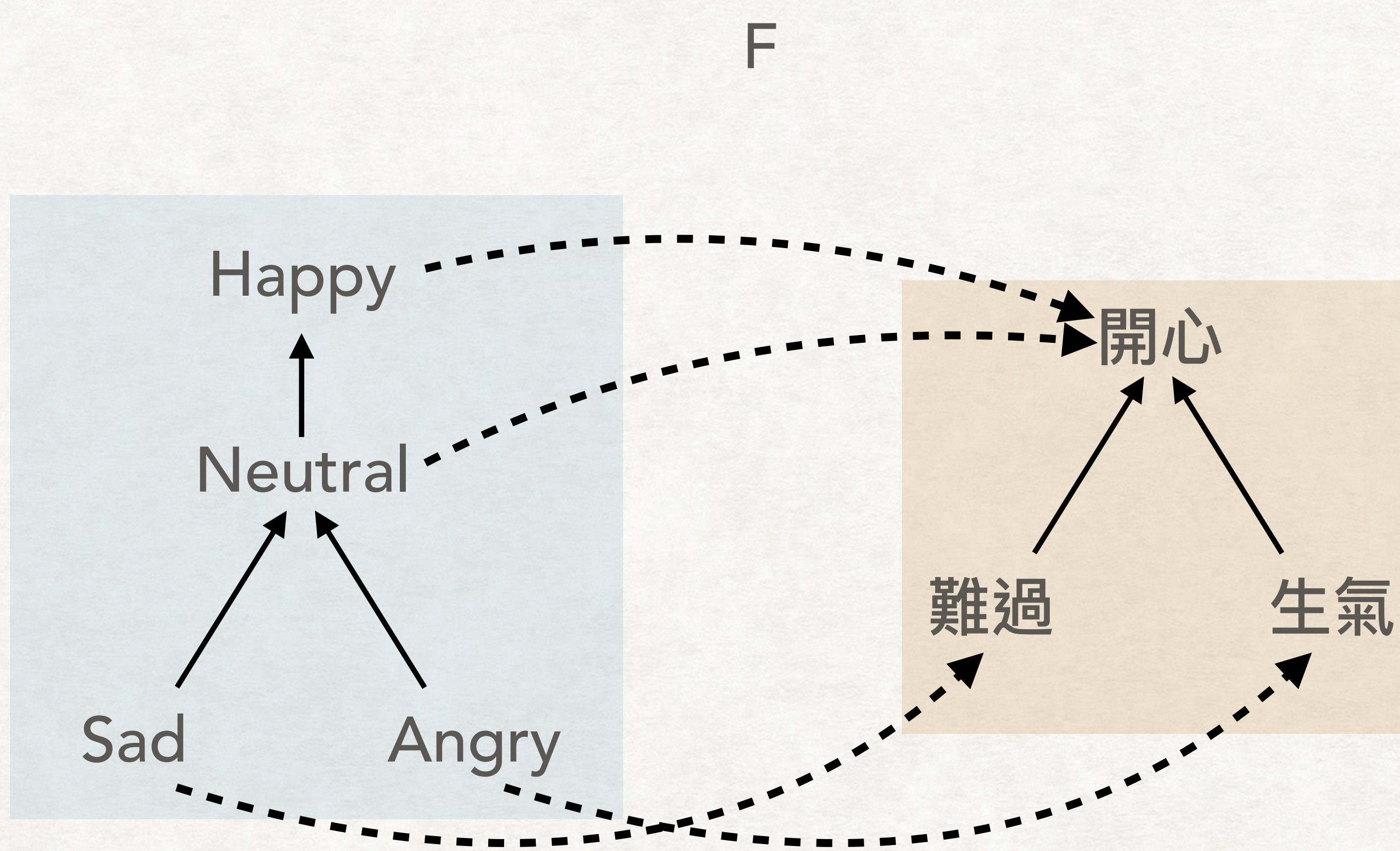


Category of
English words: E



Category of
Chinese words: C

WHAT IS A FUNCTOR? LET'S START WITH A FAMILIAR EXAMPLE

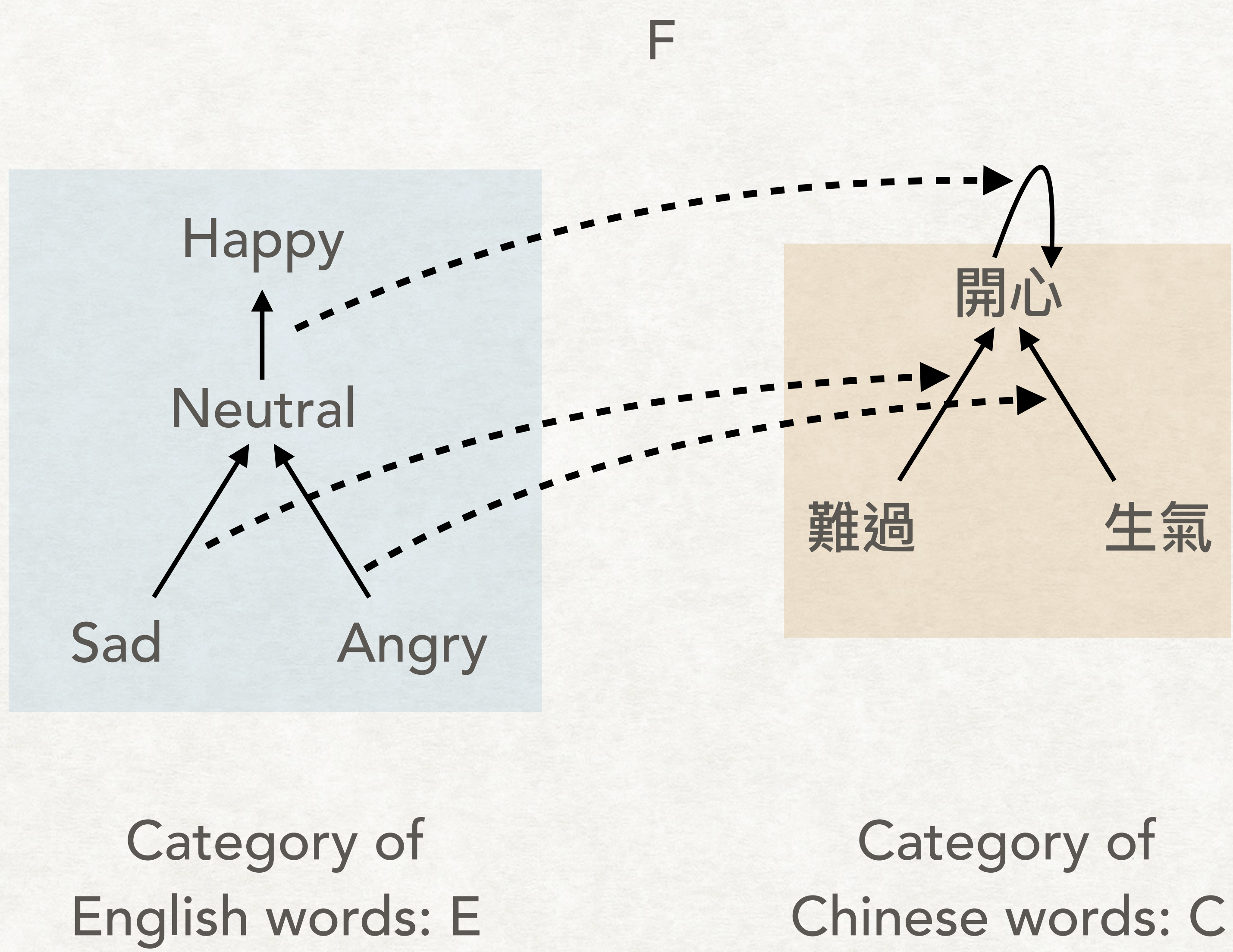


Category of
English words: E

Category of
Chinese words: C

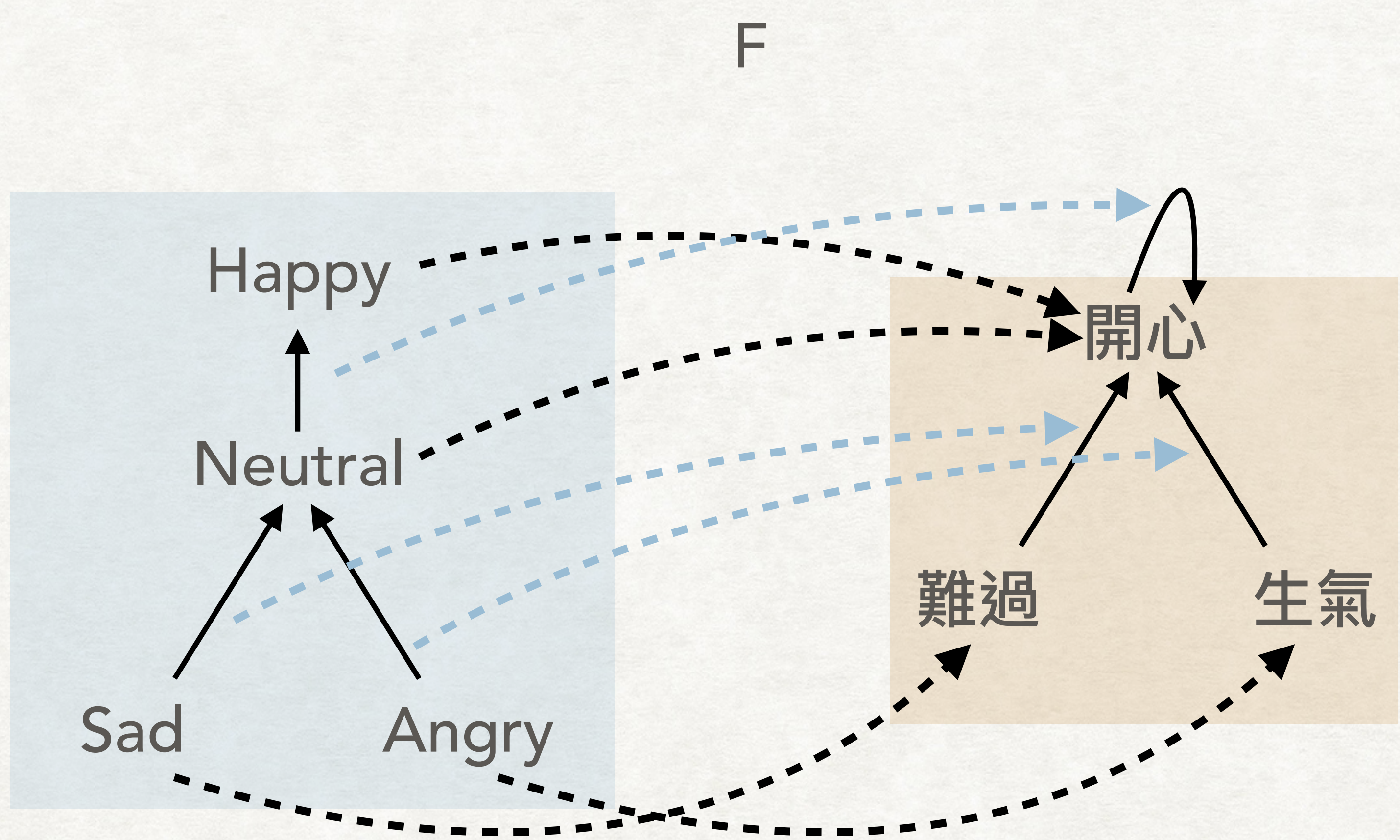
MAP EVERY OBJECT IN E TO AN OBJECT IN C

WHAT IS A FUNCTOR? LET'S START WITH A FAMILIAR EXAMPLE



MAP EVERY MORPHISM IN E TO A MORPHISM IN C

WHAT IS A FUNCTOR? LET'S START WITH A FAMILIAR EXAMPLE



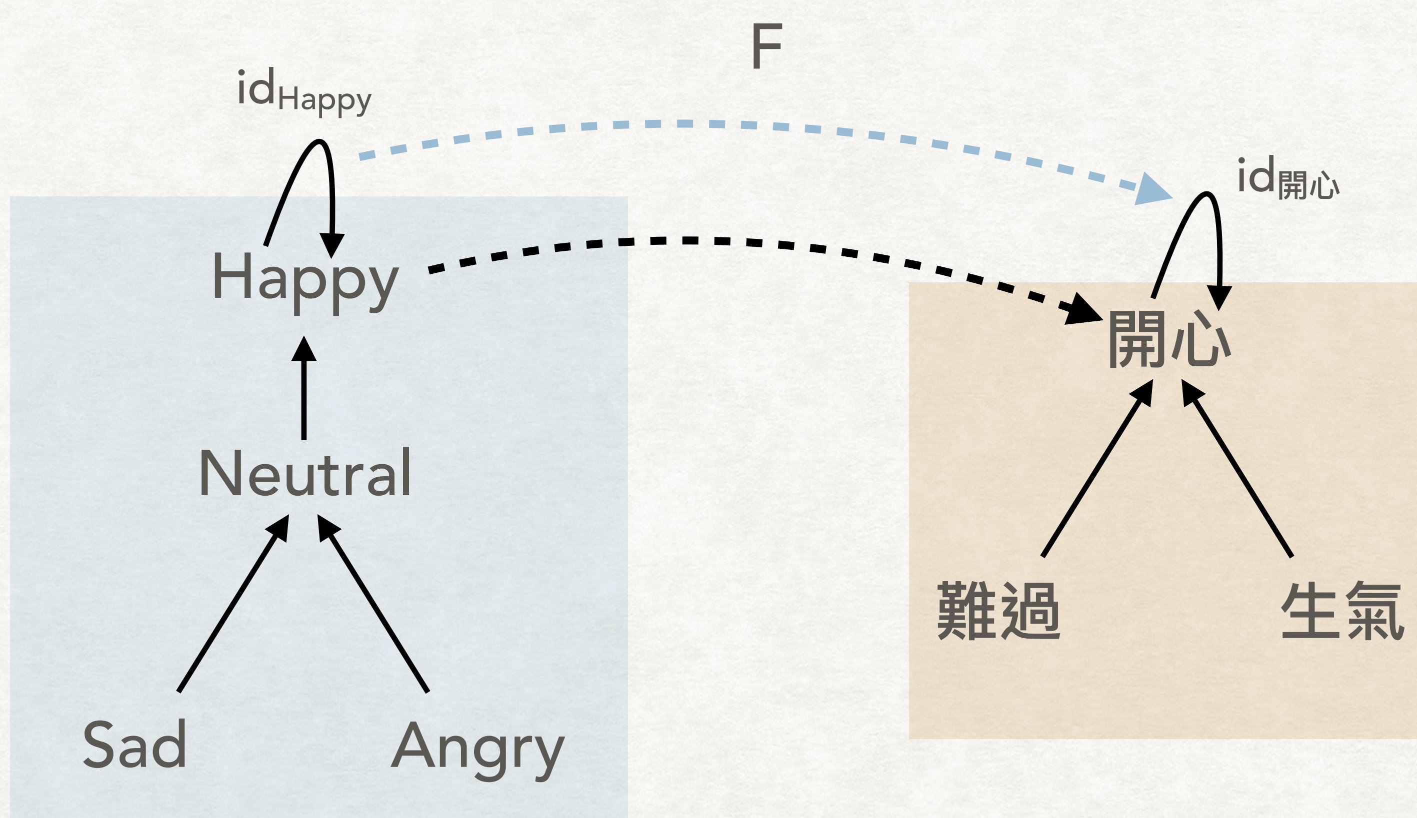
Category of
English words: E

Category of
Chinese words: C

A FUNCTOR BASICALLY MAPS EACH OBJECT/MORPHISM FROM ONE CATEGORY TO AN OBJECT/MORPHISM IN ANOTHER CATEGORY

BUT THERE ARE TWO RULES

WHAT IS A FUNCTOR? LET'S START WITH A FAMILIAR EXAMPLE



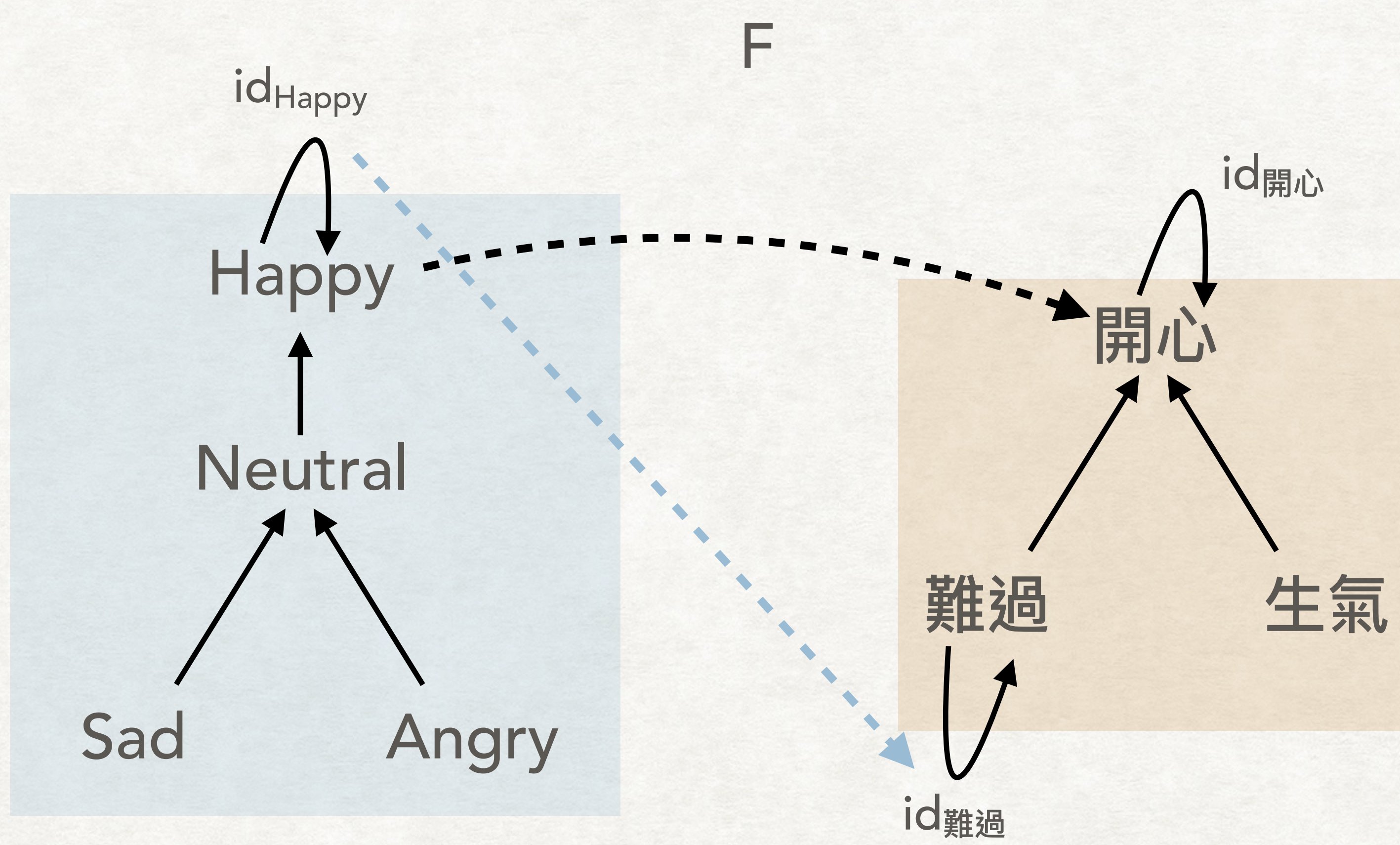
Category of
English words: E

Category of
Chinese words: C

$F(ID_{Happy}) = ID_{F(Happy)}$

Same rule applies for each object and its identity morphism

WHAT IS A FUNCTOR? LET'S START WITH A FAMILIAR EXAMPLE

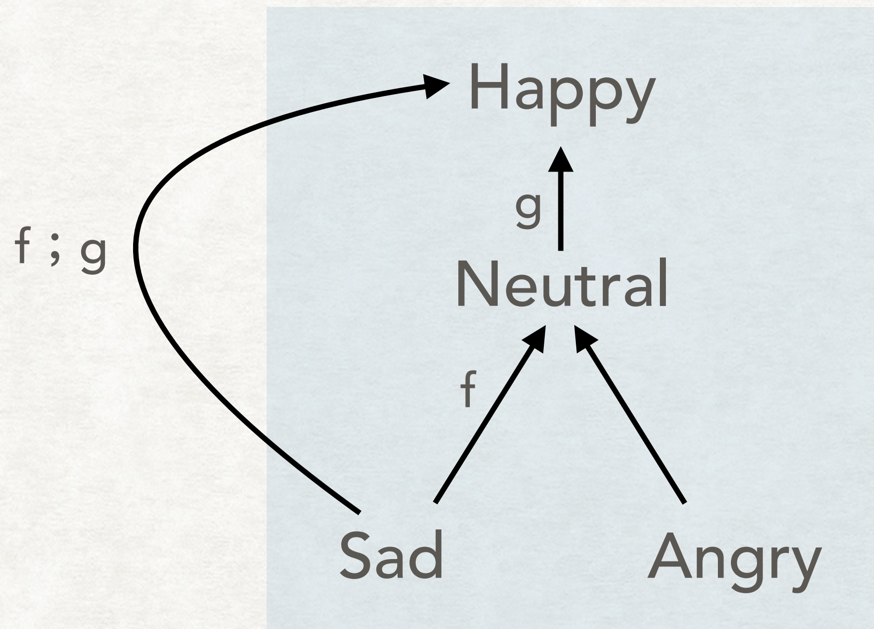


Category of
English words: E

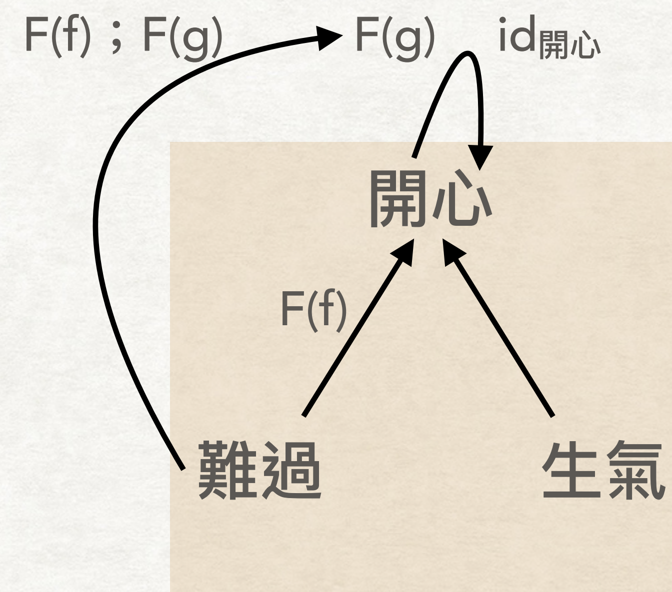
Category of
Chinese words: C

Is $F(ID_{Happy}) = ID_{F(Happy)}$?
$F(ID_{Happy}) = id_{難過}$ but
$ID_{(F(Happy))} = id_{開心}$
Hence this mapping is not a valid functor

WHAT IS A FUNCTOR? LET’S START WITH A FAMILIAR EXAMPLE



Category of
English words: E

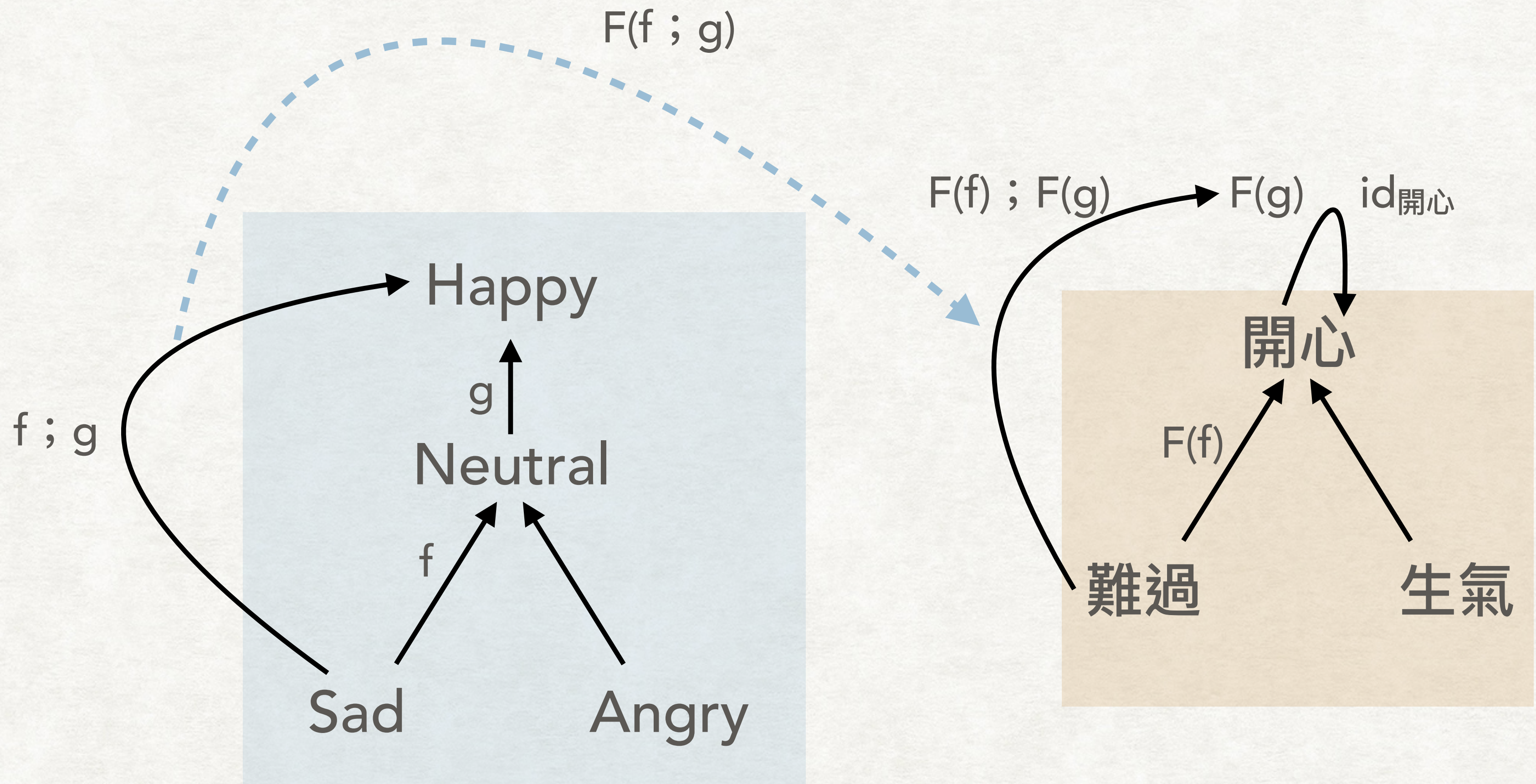


Category of
Chinese words: C

$F(f \,;\, g) = F(f) \,;\, F(g)$

Same rule applies for each object and its identity morphism

WHAT IS A FUNCTOR? LET'S START WITH A FAMILIAR EXAMPLE

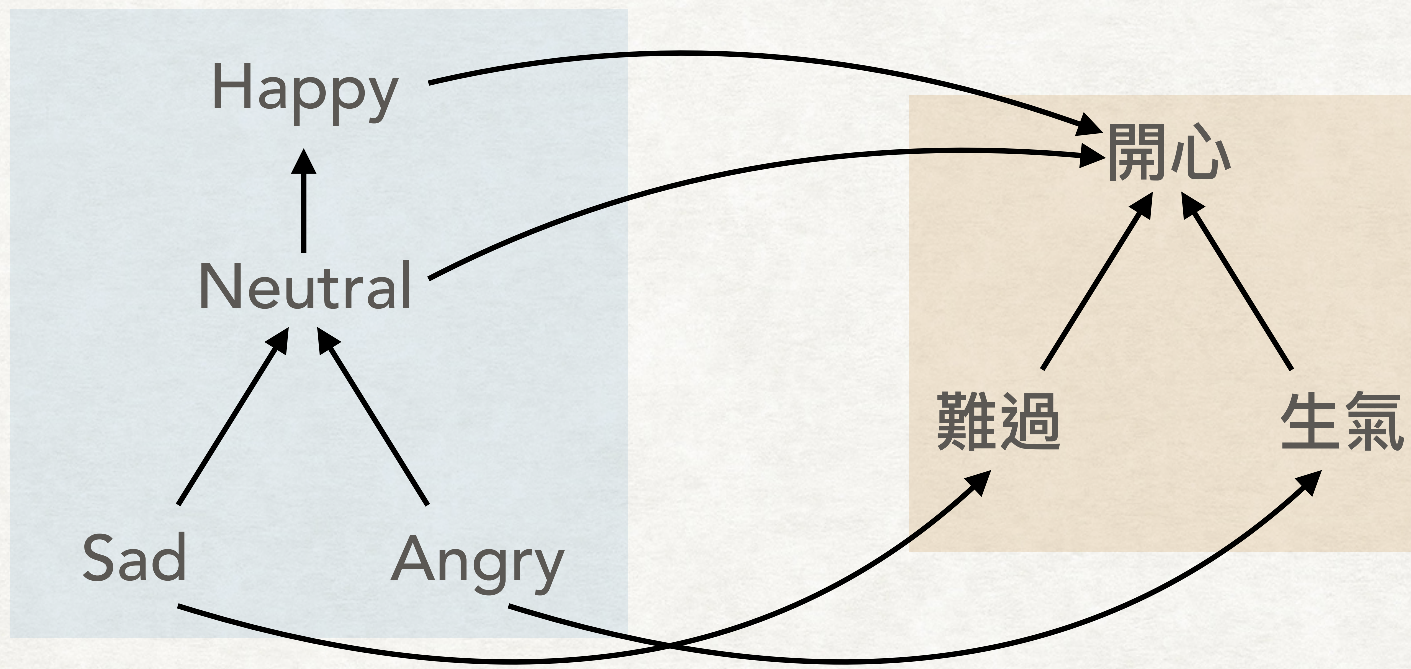


Category of
English words: E

Category of
Chinese words: C

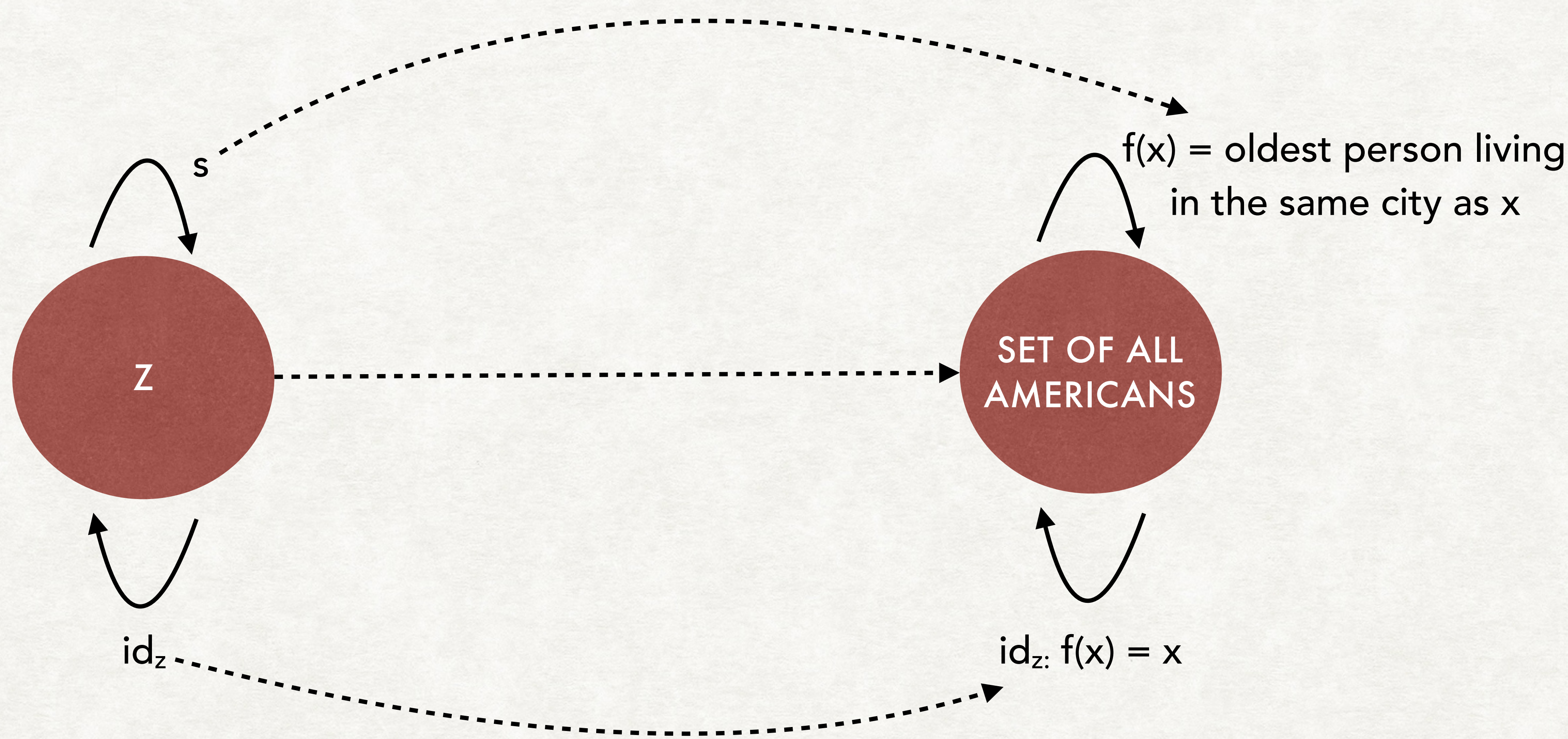
F(f ; g) = F(f) ; F(g)
Same rule applies for each object and its identity morphism

WHAT IS A FUNCTOR? LET'S START WITH A FAMILIAR EXAMPLE



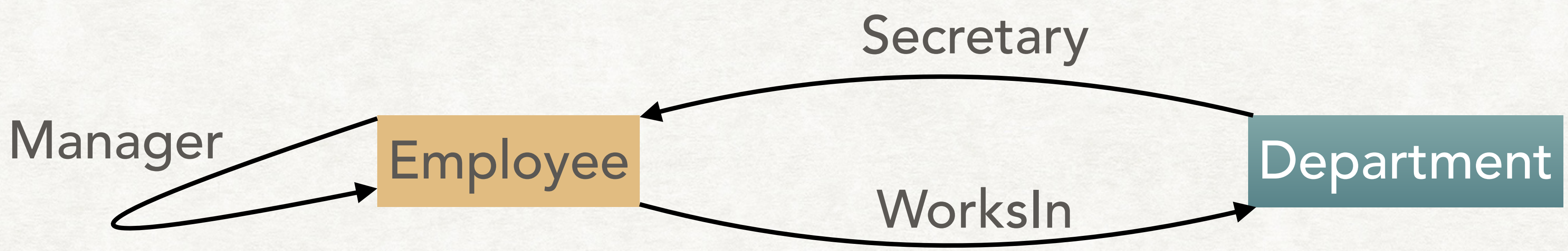
FUNCTORS BETWEEN PARTIAL ORDERS ARE MONOTONE FUNCTIONS

FUNCTOR: ONE MORE EXAMPLE

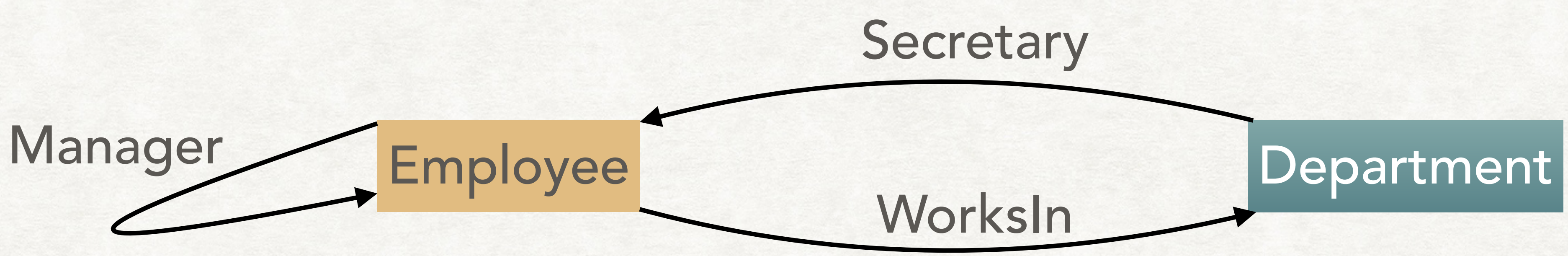


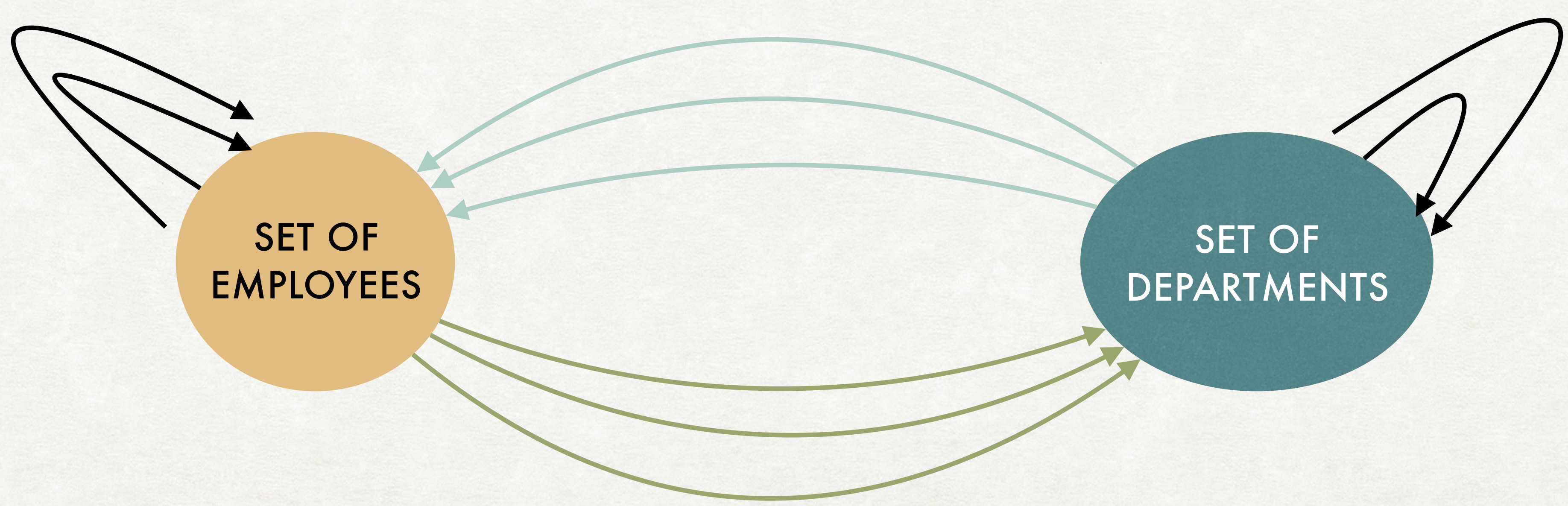
s ; s = s

CATEGORY OF DATABASE SCHEMAS

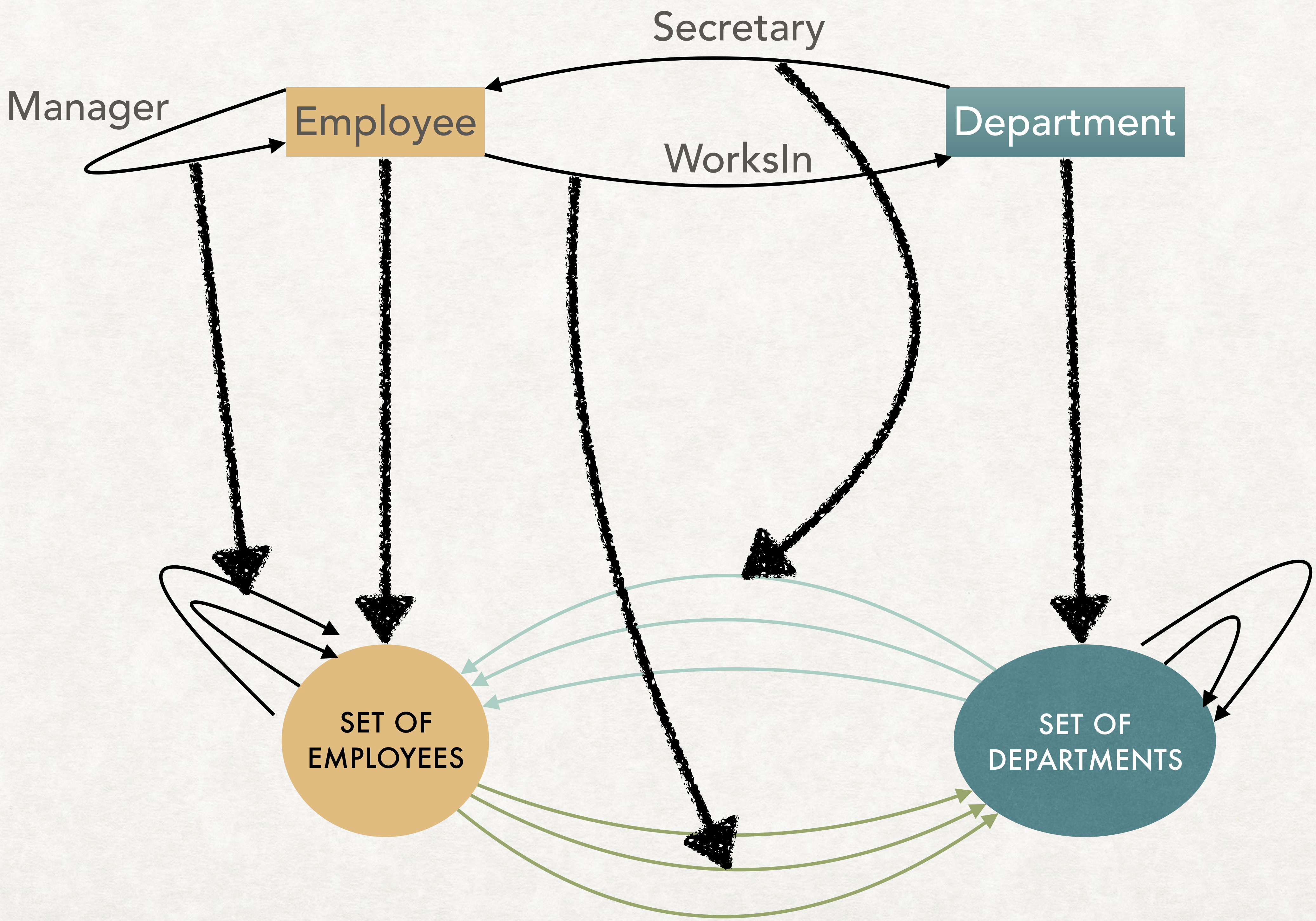


CATEGORY OF DATABASE SCHEMAS

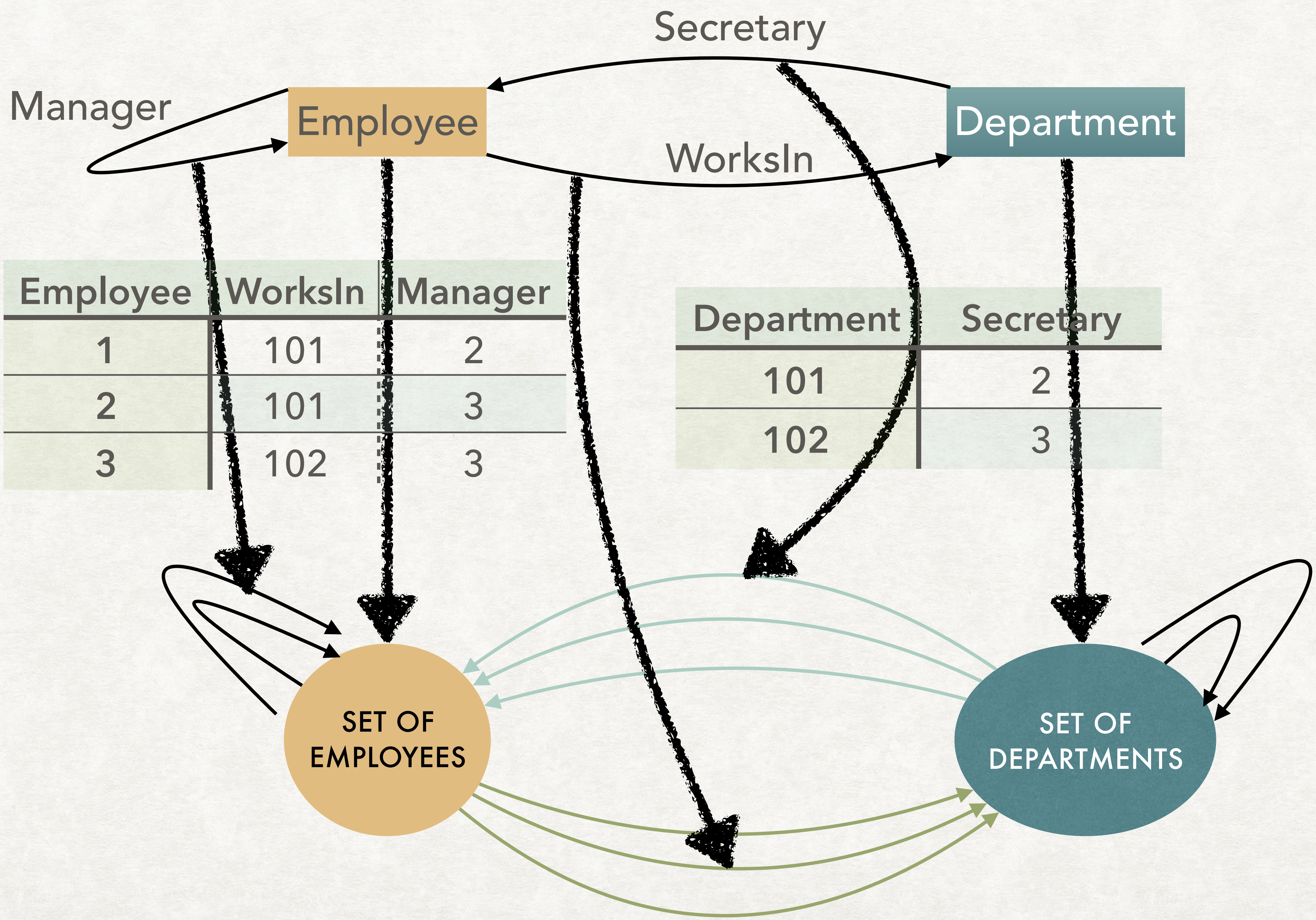




CATEGORY OF DATABASE SCHEMAS



CATEGORY OF DATABASE SCHEMAS



Employee	WorksIn	Manager
1	101	2
2	101	3
3	102	3

Department	Secretary
101	2
102	3

THANK YOU