An Introduction to Category Theory and Categorical Logic

Wolfgang Jeltsch

TTÜ Küberneetika Instituut

Teooriaseminar
April 19 and 26, 2012

From set theory to universal algebra

- classical set theory (for example, Zermelo–Fraenkel):
  - sets
  - functions from sets to sets
  - composition of functions yields function
  - identity functions exist
- adding structure and preserving it:
  - vector spaces
  - linear maps from vector spaces to vector spaces
  - composition of linear maps yields linear map
  - identity functions are linear maps
- generalization of this idea in universal algebra:
  - certain algebras with the same signature
  - homomorphisms from such algebras to other such algebras
  - composition of homomorphisms yields homomorphism
  - identity functions are homomorphisms

Beyond universal algebra

- topology based on the Kuratowski axioms:
  - topological space is a set X and a closure operator

    $$\mathrm{cl} : \mathcal{P}(X) \to \mathcal{P}(X)$$

    that fulfills certain axioms
  - continuous function from (X, cl) to (X', cl') is a function $f : X \to X'$ with

    $$f(\mathrm{cl}(A)) \subseteq \mathrm{cl}'(f(A))$$

- does not fit into the universal algebra framework:
  - closure operator operates on sets instead of single elements
  - continuity axiom uses $\subseteq$ instead of $=$
- will fit into the categorical framework

No elements anymore

- revision control system darcs:
  - repository states
  - patches that turn repository states into repository states
  - composition of patches yields patch
  - empty patches exist
- repository states do not have elements
- will fit into the categorical framework nevertheless
- more about a categorical approach to darcs in [Swierstra]

Categories

- components of a category:
  - a class of objects
  - class of morphisms, each having a unique domain and a unique codomain, which are objects
  - composition of morphisms:

$$\frac{f : A \to B \qquad g : B \to C}{gf : A \to C}$$

  - identity morphisms:

$$\mathrm{id}_A : A \to A$$

- axioms that have to hold:
  - composition is associative
  - id is left and right unit
- classes of objects and morphism are not necessarily sets:
  allows categories of sets, vector spaces, etc.
- composition is partial:
  codomain and domain must match
- above constructions lead to categories **Set**, **Vec**, etc.

Duality

- axioms still hold after doing the following:
  - swapping domain and codomain of each morphism
  - changing the argument order of composition
- opposite category $\mathcal{C}^{\mathrm{op}}$ for every category $\mathcal{C}$:
  - objects of $\mathcal{C}^{\mathrm{op}}$ are the ones of $\mathcal{C}$
  - morphisms $f : A \to B$ of $\mathcal{C}^{\mathrm{op}}$ are the morphism $f : B \to A$ of $\mathcal{C}$
  - compositions gf in $\mathcal{C}^{\mathrm{op}}$ are the compositions fg in $\mathcal{C}$
  - identities in $\mathcal{C}^{\mathrm{op}}$ are the same as in $\mathcal{C}$
- consequences:
  - for every categorical notion N, there is a dual notion N^{op} such that something is an N^{op} in $\mathcal{C}$ if it is an N in $\mathcal{C}^{\mathrm{op}}$
  - for every theorem, there is a dual theorem that refers to the dual notions

Products of categories

- product category $\mathcal{C} \times \mathcal{D}$ for any two categories $\mathcal{C}$ and $\mathcal{D}$:
  - objects
    $$(A, B)$$
    where A is an object of $\mathcal{C}$, and B is an object of $\mathcal{D}$
  - morphisms
    $$(f, g) : (A, B) \to (A', B')$$
    where $f : A \to A'$ and $g : B \to B'$
  - compositions and identities defined componentwise:
    $$(f', g')(f, g) = (f'f, g'g)$$
    $$\mathrm{id}_{(A,B)} = (\mathrm{id}_A, \mathrm{id}_B)$$
- neutral element is the category 1:
  - exactly one object
  - exactly one morphism (the identity of that object)

Categories and elements

- in general, no notion of element of an object
- however, elements can be recovered for specific kinds of categories
- furthermore, some concepts that seem to require the notion of element actually do not

Injectivity

Definition (Injectivity)

A function $f : A \to B$ is injective if and only if

$$\forall x_1, x_2 \in A \, . \, f(x_1) = f(x_2) \Rightarrow x_1 = x_2 \quad .$$

Theorem

A function $f : A \to B$ is injective if and only if

$$\forall C \, . \, \forall g_1, g_2 : C \to A \, . \, f g_1 = f g_2 \Rightarrow g_1 = g_2 \quad .$$

- above definition relies on the notion of element
- theorem gives us another property for defining injectivity:
  - does not mention elements, but only sets and functions (point-free style)
  - can therefore be generalized to arbitrary categories
  - leads to the notion of monomorphism

Surjectivity

Definition (Surjectivity)

A function $f : A \to B$ is surjective if and only if

$$\forall y \in B \,.\, \exists x \in A \,.\, f(x) = y \quad .$$

Theorem

A function $f : A \to B$ is surjective if and only if

$$\forall C \,.\, \forall g_1, g_2 : B \to C \,.\, g_1 f = g_2 f \Rightarrow g_1 = g_2 \quad .$$

- theorem gives us point-free definition
- generalization to arbitrary categories leads to the notion of epimorphism
- point-free style makes it clear that monomorphism and epimorphism (injectivity and surjectivity) are duals

Isomorphisms

- generalization of bijections
- morphism $f : A \to B$ is an isomorphism if there is an $f^{-1} : B \to A$ such that

$$f^{-1}f = \mathrm{id}_A \qquad ff^{-1} = \mathrm{id}_B$$

- objects A and B are isomorphic ($A \cong B$) if there exists an isomorphism $f : A \to B$

Cartesian products

- pair construction:

$$\frac{x \in A \qquad y \in B}{(x, y) \in A \times B}$$

- pair destruction:

$$\pi_1 : A \times B \to A \qquad \pi_2 : A \times B \to B$$

- destruction is point-free, construction is not
- construction can be made point-free:

$$\frac{f : C \to A \qquad g : C \to B}{\langle f, g \rangle : C \to A \times B}$$

where

$$\forall z \in C \,.\, \langle f, g \rangle (z) = (f(z), g(z))$$

Products

- generalization of cartesian products
- a product of A and B is an object $A \times B$ together with morphisms

$$\pi_1 : A \times B \to A \qquad \pi_2 : A \times B \to B$$

  (called projections) for which the following holds:
  - for every object C, we have

$$\frac{f : C \to A \qquad g : C \to B}{\langle f, g \rangle : C \to A \times B}$$

  - the following holds:

$$\pi_1 \langle f, g \rangle = f \qquad \pi_2 \langle f, g \rangle = g$$

  - the morphism $\langle f, g \rangle$ is unique
- two objects A and B may not have a product
- products of two specific objects are unique up to isomorphism

Coproducts

- duals of products
- a coproduct of A and B is an object $A + B$ together with morphisms

$$\iota_1 : A \to A + B \qquad \iota_2 : B \to A + B$$

(called injections) for which the following holds:

  - for every object C, we have

$$\frac{f : A \to C \qquad g : B \to C}{[f, g] : A + B \to C}$$

  - the following holds:

$$[f, g]\iota_1 = f \qquad [f, g]\iota_2 = g$$

  - the morphism $[f, g]$ is unique
- two objects A and B may not have a coproduct
- coproducts of two specific objects are unique up to isomorphism

Terminal and initial objects

- nullary versions of products and coproducts
- 1 is a terminal object if there is a unique morphism

$$! : C \to 1$$

  for every object C
- 0 is an initial object if there is a unique morphism

$$? : 0 \to C$$

  for every object C
- terminal and initial objects are unique up to isomorphism
- if terminal object exists, $A \times 1$ and $1 \times A$ exist for every object A, and we have

$$A \times 1 \cong A \cong 1 \times A$$

- analogously for initial object

Function spaces

- for sets A and B, we have

$$B^A = \{f \mid f : A \to B\}$$

- Currying:

$$\frac{f : C \times A \to B}{\lambda_f : C \to B^A}$$

- function application:

$$\epsilon : B^A \times A \to B$$

where

$$\epsilon(f, x) = f(x)$$

Exponentials

- generalization of function spaces
- defined for categories where all (binary) products exist
- an exponential of A and B is an object B^A together with a morphism

$$\epsilon : B^A \times A \to B$$

  for which the following holds:
  - for every object C, we have

$$\frac{f : C \times A \to B}{\lambda_f : C \to B^A}$$

  - the following holds:

$$\epsilon \langle \lambda_f \pi_1, \pi_2 \rangle = f$$

  - the morphism λ_f is unique
- two objects A and B may not have an exponential
- exponentials of two specific objects are unique up to isomorphism

Cartesian closed categories and beyond

Definition (Cartesian closed category)

A category is a cartesian closed category (CCC) if it has all (binary) products, a terminal object, and all exponentials.

Definition (Bicartesian closed category)

A category is a bicartesian closed category (BCCC), sometimes called cocartesian closed category (CCCC), if it is a CCC and has all (binary) coproducts and an initial object.

Categorical logic basics

- categories as models of logics
- general idea:
  - objects model propositions
  - if objects A and B model propositions φ and ψ, morphisms $f : A \to B$ model proofs of $\varphi \vdash \psi$
  - composition models composition of proofs:

$$\frac{\varphi \vdash \psi \qquad \psi \vdash \chi}{\varphi \vdash \chi}$$

  - identities model identity rule:

$$\frac{}{\varphi \vdash \varphi}$$

- BCCCs are the models of intuitionistic propositional logic
- for modeling other intuitionistic logics, extend BCCCs with additional structure
- even linear logic can be modeled by extended BCCCs

Products and conjunctions

- $\times$ models $\wedge$:
  - $\langle\cdot,\cdot\rangle$ proves conjunction introduction:

$$\frac{\chi \vdash \varphi \qquad \chi \vdash \psi}{\chi \vdash \varphi \wedge \psi}$$

  - projections prove conjunction elimination:

$$\frac{}{\varphi \wedge \psi \vdash \varphi} \qquad \frac{}{\varphi \wedge \psi \vdash \psi}$$

- 1 models $\top$:
  - ! proves truth:

$$\frac{}{\chi \vdash \top}$$

Coproducts and disjunctions

- $+$ models $\vee$:
  - $[\cdot, \cdot]$ proves disjunction elimination:

$$\frac{\varphi \vdash \chi \qquad \psi \vdash \chi}{\varphi \vee \psi \vdash \chi}$$

  - injections prove disjunction introduction:

$$\frac{}{\varphi \vdash \varphi \vee \psi} \qquad \frac{}{\psi \vdash \varphi \vee \psi}$$

- 0 models $\bot$:
  - ? proves *ex falso quodlibet*:

$$\frac{}{\bot \vdash \chi}$$

Exponentials and implications

- exponentiation models $\Rightarrow$:
  - λ proves implication introduction:

$$\frac{\chi \wedge \varphi \vdash \psi}{\chi \vdash \varphi \Rightarrow \psi}$$

  - ϵ proves implication elimination:

$$\frac{}{(\varphi \Rightarrow \psi) \wedge \varphi \vdash \psi}$$

Functors

- structure-preserving maps between categories ("category homomorphisms")
- functor $F : \mathcal{C} \to \mathcal{D}$ actually consists of two maps:
  - a map from objects of $\mathcal{C}$ to objects of $\mathcal{D}$
  - a map from morphisms of $\mathcal{C}$ to morphisms of $\mathcal{D}$
- notation for application of these maps
  uses juxtaposition of functor and argument:
  - application of F's object map to object A is FA
  - application of F's morphism map to morphism f is Ff
- axioms:
  - transformation of domains and codomains:

$$\frac{f : A \to B}{Ff : FA \to FB}$$

  - compatibility with composition:

$$F(gf) = (Fg)(Ff)$$

  - compatibility with identities:

$$F\mathrm{id}_A = \mathrm{id}_{FA}$$

Functor examples

- power set functor from **Set** to itself:
  - object map turns sets into their power sets:

$$\mathcal{P}A = \{M \mid M \subseteq A\}$$

  - morphism map turns functions into elementwise applications of them:

$$(\mathcal{P}f)(M) = \{f(x) \mid x \in M\}$$

- list functor from functional programming is similar:
  - object map turns element types into list types
  - morphism map turns functions into elementwise applications of them
- projections of product categories:
  - object maps turn pairs of objects into objects:

$$\Pi_1(A, B) = A \qquad \Pi_2(A, B) = B$$

  - morphism maps turn pairs of morphisms into morphisms:

$$\Pi_1(f, g) = f \qquad \Pi_2(f, g) = g$$

Functor intuitions

- container intuition:
  - object map turns types/sets of elements into types/sets of containers
  - morphism map turns functions into elementwise applications of them
- effect intuition:
  - object map turns types/sets of results into types/sets of effectful computations
  - morphism map turns functions into functions that append the former functions to effectful computations
- application to power set functor:
  - sets are containers
  - sets denote nondeterministic computations

Category of small categories

- composition $GF : \mathcal{C} \to \mathcal{E}$ of functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{E}$:
  - object map is composition of object maps of F and G
  - morphism map is composition of morphism maps of F and G
- identity functor $\mathrm{Id} : \mathcal{C} \to \mathcal{C}$:
  - object map is identity function on objects
  - morphism map is identity function on morphism
- category **Cat** of categories and functors:
  - objects are all categories
  - morphisms $F : \mathcal{C} \to \mathcal{D}$ are the functors $F : \mathcal{C} \to \mathcal{D}$
  - composition is functor composition
  - identities are the identity functors
- set theory in use might not allow for the class of all categories
- objects of **Cat** are only all small categories:
  - object classes are sets
  - morhpism classes are sets

Natural transformations

- natural transformation τ from functor F to functor G is an indexed family of morphisms

$$\tau_A : FA \to GA \ ,$$

one for each object A
- compatibility with morphism maps:

$$\tau_B(Ff) = (Gf)\tau_A$$

Functor categories

- important properties of natural transformations:
  - pointwise composition yields natural transformation
  - pointwise identities are natural transformations
- functor category:
  - objects are all functors from a certain source to a certain target category
  - morphisms $\tau : F \to G$ are the natural transformations from F to G
  - compositions and identities constructed pointwise
- natural isomorphisms:
  - are the isomorphisms of functor categories
  - are exactly those natural transformations that consist only of isomorphisms

Revisiting products and coproducts

- products:
  - $\times$ is a functor from $\mathcal{C} \times \mathcal{C}$ to $\mathcal{C}$ with
    $$f \times g = \langle f\pi_1, g\pi_2 \rangle$$
  - π_1 and π_2 are natural transformations:
    $$\pi_1 : \times \to \Pi_1 \qquad \pi_2 : \times \to \Pi_2$$
- coproducts:
  - $+$ is a functor from $\mathcal{C} \times \mathcal{C}$ to $\mathcal{C}$ with
    $$f + g = [\iota_1 f, \iota_2 g]$$
  - ι_1 and ι_2 are natural transformations:
    $$\iota_1 : \Pi_1 \to + \qquad \iota_2 : \Pi_2 \to +$$

Revisiting exponentials

- exponentiation is a functor $E : \mathcal{C}^{\text{op}} \times \mathcal{C} \to \mathcal{C}$ with

$$g^f = \lambda_{g\epsilon(\text{id}_{B^A} \times f)}$$

- ϵ is a natural transformation

$$\epsilon : E \times \Pi_1 \to \Pi_2 \quad ,$$

where $E \times \Pi_1$ is the functor with

$$(E \times \Pi_1)A = EA \times \Pi_1 A$$

and

$$(E \times \Pi_1)f = Ef \times \Pi_1 f$$

Monoidal categories

- categories $\mathcal{C}$ that have the following additional structure:
  - a functor
    $$\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C} \enspace ,$$
    called the tensor product
  - an object I, called the unit object
  - a natural isomorphism α establishing associativity of $\otimes$:
    $$\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$$
  - two natural isomorphisms λ and ρ establishing the fact that I is a left and right unit of $\otimes$:
    $$\lambda_A : I \otimes A \to A \qquad \rho_A : A \otimes I \to A$$
- axiom:

  For any objects A and B, all morphisms from A to B that are built solely from $\otimes$, α, λ, and ρ are equal.
- three dedicated equalities actually enough, since the rest follows from Mac Lane's coherence theorem

Monoidal functors

- monoidal functor F from monoidal category $(\mathcal{C}, \otimes, I)$ to monoidal category $(\mathcal{C}', \otimes', I')$ consists of the following:
  - a functor from $\mathcal{C}$ to $\mathcal{C}'$ (also named F)
  - two natural transformations m and n, called coherence maps:

$$m_{A,B} : FA \otimes' FB \to F(A \otimes B)$$
$$n : I' \to FI$$

- axioms ensure compatibility of coherence maps with α, λ, and ρ
- F is called strong if coherence maps are isomorphisms
- comonoidal functor is dual of monoidal functor (coherence maps go into opposite direction)

Monoidal category and functor examples

- monoidal category examples:
  - if $\mathcal{C}$ has all finite products, $(\mathcal{C}, \times, 1)$ is a monoidal category
  - if $\mathcal{C}$ has all finite coproducts, $(\mathcal{C}, +, 0)$ is a monoidal category
- monoidal functor examples:
  - list functor is a monoidal functor from $(\mathcal{C}, \times, 1)$ to itself:
    - m corresponds to *uncurry zip* in Haskell
    - n corresponds to *repeat* in Haskell
  - if $\mathcal{C}$ is a category with all finite products and $F : \mathcal{C} \to \mathcal{C}$, then F is a comonoidal functor from $(\mathcal{C}, \times, 1)$ to itself:

$$m = \langle F\pi_1, F\pi_2 \rangle \qquad\qquad n = !_{F1}$$

  - infinite list functor is a strong monoidal functor from $(\mathcal{C}, \times, 1)$ to itself:
    - coherence maps as for lists
    - inverses of coherence maps are the coherence maps of the abovementioned comonoidal functor

Monads

- monad on a category $\mathcal{C}$ consists of the following:
  - a functor $T : \mathcal{C} \to \mathcal{C}$
  - two natural transformations

$$\eta : \mathrm{Id} \to T \qquad\qquad \mu : TT \to T$$

- axioms:

$$\mu_A(T\mu_A) = \mu_A\mu_{TA} : TTTA \to TA$$
$$1_{TA} = \mu_A(T\eta_A) = \mu_A\eta_{TA} : TA \to TA$$

- consequences:
  - For every n, there are natural transformations from T^n to T that are built solely from T, η, and μ.
  - For every n, all such transformations are equal.

Monad examples

- power set monad:
  - μ is general union

$$\bigcup : \mathcal{P}(\mathcal{P}A) \to \mathcal{P}A$$

  - η is singleton construction

$$\{\cdot\} : A \to \mathcal{P}A$$

- list monad:
  - μ corresponds to *concat* in Haskell
  - η corresponds to $\lambda x \to [x]$ in Haskell

Monad intuitions

- container that can be built from arbitrarily nested containers:
  - μ turns a two-level nested container into a flat container
  - η turns a single value (zero-level nested container) into a singleton container
- effectful computations that can be built from sequences of computations:
  - μ turns a sequence of two computations into a single computation
  - η turns a result value into a computation without effect that just returns this value

Comonads

- duals of monads
- comonad on a category $\mathcal{C}$ consists of the following:
  - a functor $U : \mathcal{C} \to \mathcal{C}$
  - two natural transformations

$$\varepsilon : U \to \mathrm{Id} \qquad \delta : U \to UU$$

- axioms:

$$(U\delta_A)\delta_A = \delta_{UA}\delta_A : UA \to UUUA$$
$$1_{UA} = (U\varepsilon_A)\delta_A = \varepsilon_{UA}\delta_A : UA \to UA$$

- consequences:
  - For every n, there are natural transformations from U to U^n that are built solely from U, ε, and δ.
  - For every n, all such transformations are equal.

Comonad example and intuition

- infinite list comonad as an example:
  - δ corresponds to *tails* in Haskell
  - ε corresponds to *head* in Haskell
- intuition is that of containers that can be turned into arbitrarily nested containers:
  - δ turns a flat container into a two-level nested container
  - ε turns a flat container into a single value, which is taken from a special position inside the container

Kleisli and Co-Kleisli categories

- Kleisli category of a monad (T, η, μ) on a category $\mathcal{C}$:
  - objects of the Kleisli category are the objects of $\mathcal{C}$
  - morphisms $f : A \to B$ of the Kleisli category are the morphisms $f : A \to TB$ of $\mathcal{C}$
  - compositions gf in the Kleisli category correspond to morphisms $\mu(Tg)f$ in $\mathcal{C}$
  - identities in the Kleisli category correspond to η in $\mathcal{C}$
- Kleisli category intuition:

  morphisms are effectful computations
  that also have an input
- Co-Kleisli categories are the duals of Kleisli categories

References

Andrea Asperti and Giuseppe Longo
Categories, Types, and Structures
http://www.cs.unibo.it/~asperti/PAPERS/book.pdf

Steve Awodey
Category Theory
http://www.andrew.cmu.edu/course/80-413-713/notes/cats.pdf

Samson Abramsky and Nikos Tzevelekos
Introduction to Categories and Categorical Logic
http://arxiv.org/pdf/1102.1313v1

Wouter Swierstra
Theory of patches
http://sneezy.cs.nott.ac.uk/fplunch/weblog/?p=4