

Topos Theory

Archived study guides

These archived materials are no longer updated. This archive was exported from a saved copy of these guides.

Export date: 2026-10-05T19:26:30.244Z

Contents

Unit 1 – Introduction to Category Theory - 4 guides

Unit 2 – Functors and Natural Transformations - 3 guides

Unit 3 – Limits and Colimits - 4 guides

Unit 4 – Adjoint Functors - 3 guides

Unit 5 – Cartesian Closed Categories - 3 guides

Unit 6 – Subobject Classifiers and Topoi - 3 guides

Unit 7 – Elementary Topoi and Examples - 3 guides

Unit 8 – Sheaves and Sheafification - 3 guides

Unit 9 – Grothendieck Topoi - 3 guides

Unit 10 – Geometric Morphisms and Classifying Topoi - 3 guides

Unit 11 – Internal Logic and Mitchell–Bénabou Language - 3 guides

Unit 12 – Topos Theory: Algebraic and Geometric Views - 3 guides

Unit 13 – Topos Theory and Foundations of Mathematics - 3 guides

Unit 14 – Topos Theory: Advanced Applications - 3 guides

Unit 1 – Introduction to Category Theory

1.1 Basic definitions and examples of categories

Categories are the building blocks of abstract mathematics. They consist of objects and morphisms, which represent relationships between objects. Understanding these components is crucial for grasping more complex concepts in category theory.

Categories appear across various mathematical fields. From sets and functions to groups and homomorphisms, these structures provide a unified framework for studying different mathematical concepts. Composition and identity morphisms play key roles in defining category properties.

Fundamental Concepts of Categories

Components of categories

- Category encompasses abstract mathematical structure composed of objects and morphisms (arrows) representing relationships between objects
- Objects form fundamental elements of a category visually depicted as dots or points in diagrams
- Morphisms (arrows) connect objects within the category graphically represented as directed arrows between objects
- Key properties include composition of morphisms allowing combination of arrows and identity morphisms existing for each object

Examples across mathematics

- Category of Sets (Set) contains sets as objects and functions between sets as morphisms (maps between elements)
- Category of Groups (Grp) comprises groups as objects and group homomorphisms as morphisms (structure-preserving maps)
- Category of Topological Spaces (Top) consists of topological spaces as objects and continuous functions as morphisms (preserving open sets)
- Category of Vector Spaces (Vect) includes vector spaces over a field as objects and linear transformations as morphisms (preserving vector operations)
- Category of Rings (Ring) encompasses rings as objects and ring homomorphisms as morphisms (preserving algebraic structure)

Composition of morphisms

- Composition of morphisms combines two morphisms into a single arrow denoted by $g \circ f$ or gf
- For morphisms $f: A \rightarrow B$ and $g: B \rightarrow C$, their composition $g \circ f: A \rightarrow C$ maps elements from A to C
- Associativity property ensures $(h \circ g) \circ f = h \circ (g \circ f)$ allowing omission of parentheses in compositions
- Composition diagrams visually represent morphism compositions with commutative diagrams showing equivalent paths between objects

Role of identity morphisms

- Identity morphism exists for each object in a category denoted as 1_A or id_A for object A
- Properties include $f \circ 1_A = f$ and $1_B \circ f = f$ for any morphism $f: A \rightarrow B$
- Identity morphisms serve as neutral elements for composition allowing objects to be treated as morphisms
- Enable definition of isomorphisms and other crucial concepts in category theory

1.2 Morphisms, isomorphisms, and functors

Category Theory fundamentals are crucial building blocks for understanding more complex concepts in Topos Theory. This section introduces key morphisms like monomorphisms, epimorphisms, and isomorphisms, which generalize familiar concepts from set theory to broader categorical settings.

Functors, the focus of the next part, are essential tools for connecting different categories. They allow us to map objects and morphisms between categories while preserving important structural relationships, forming the basis for more advanced categorical constructions.

Category Theory Fundamentals

Monomorphisms, epimorphisms, and isomorphisms

- Monomorphism
- Morphism $f: A \rightarrow B$ in category C acts like injective function generalizes injectivity from Set theory
- Left-cancellative property ensures for morphisms $g, h: X \rightarrow A$, $f \circ g = f \circ h$ implies $g = h$
- Examples: inclusion maps (subsets), group homomorphisms with trivial kernel
- Epimorphism
- Morphism $f: A \rightarrow B$ in category C behaves like surjective function extends surjectivity concept
- Right-cancellative property guarantees for morphisms $g, h: B \rightarrow Y$, $g \circ f = h \circ f$ implies $g = h$
- Examples: quotient maps (groups), projection maps (product spaces)
- Isomorphism
- Morphism $f: A \rightarrow B$ in category C has inverse $g: B \rightarrow A$ satisfying $g \circ f = 1_A$ and $f \circ g = 1_B$
- Generalizes bijective functions preserves structure between objects
- Examples: vector space isomorphisms, group isomorphisms, homeomorphisms (topology)

Isomorphisms and two-sided inverses

- Proof outline 1. Assume f is isomorphism
- Definition provides g with $g \circ f = 1_A$ and $f \circ g = 1_B$
- g serves as two-sided inverse of f 2. Assume f has two-sided inverse g
- Given $g \circ f = 1_A$ and $f \circ g = 1_B$

- Conditions match isomorphism definition
- Key points
- Inverse uniqueness ensures only one two-sided inverse exists
- Isomorphisms possess both monomorphism and epimorphism properties
- Examples: matrix inverses, function inverses (bijective functions)

Functors and Their Properties

Functors between categories

- Functor definition
- Map $F : C \rightarrow D$ between categories C and D preserves categorical structure
- Assigns objects to objects and morphisms to morphisms maintaining relationships
- Components of functor
- Object assignment maps each object A in C to $F(A)$ in D
- Morphism assignment takes each morphism f in C to $F(f)$ in D
- Examples of functors
- Forgetful functor from Group to Set strips group structure
- Power set functor from Set to Set maps sets to their power sets
- Fundamental group functor from Top to Group associates topological spaces with groups
- Constant functor maps all objects to single object and all morphisms to identity
- Identity functor maps category to itself preserving all structure

Properties of functors

- Composition preservation
- For composable morphisms $f : A \rightarrow B$ and $g : B \rightarrow C$ in C
- Ensures $F(g \circ f) = F(g) \circ F(f)$ maintaining operation structure
- Identity morphism preservation
- For object A in C , functor maps $F(1_A) = 1_{F(A)}$
- Preserves identity elements across categories
- Consequences of properties
- Commutative diagrams remain commutative when mapped by functors
- Isomorphisms transform into isomorphisms under functor application
- Types of functors
- Covariant functors preserve morphism direction (standard functors)

- Contravariant functors reverse morphism direction (dual category relationship)
- Functors and category structure
- Maintain source and target object relationships for morphisms
- Map domains to domains and codomains to codomains preserving overall structure

1.3 Duality and opposite categories

Opposite categories flip the direction of morphisms, creating a mirror image of the original category. This concept introduces duality, allowing us to explore relationships between seemingly different structures and simplify proofs.

Dual constructions like products and coproducts, equalizers and coequalizers, and limits and colimits showcase the power of duality. By reversing arrows, we gain new insights into categorical structures and their connections.

Opposite Categories and Duality

Definition of opposite categories

- Opposite category (dual category) construction reverses morphism directions
- Objects remain identical to original category
- Morphisms $f : A \rightarrow B$ become $f^{op} : B \rightarrow A$ in opposite category
- Composition in opposite category reverses order
- Original: $f : A \rightarrow B$ and $g : B \rightarrow C$
- Opposite: $g^{op} \circ f^{op} = (f \circ g)^{op}$
- Notation C^{op} represents opposite category of C

Concept of duality in categories

- Duality principle generates dual counterparts by reversing arrows and compositions
- Enhances understanding of categorical structures through symmetry
- Dual notions illuminate relationships (initial object $\leftrightarrow$ terminal object, product $\leftrightarrow$ coproduct)
- Applications include simplifying proofs and connecting concepts (monomorphism $\leftrightarrow$ epimorphism)

Examples of dual constructions

- Product vs Coproduct:
- Product: P with projections $p_1 : P \rightarrow A, p_2 : P \rightarrow B$
- Coproduct: Q with injections $i_1 : A \rightarrow Q, i_2 : B \rightarrow Q$
- Equalizer vs Coequalizer:
- Equalizer: $E \rightarrow A$ satisfies $f \circ e = g \circ e$
- Coequalizer: $A \rightarrow C$ satisfies $c \circ f = c \circ g$

- Monomorphism vs Epimorphism:
- Monomorphism: $f: A \rightarrow B$ monic if $f \circ g = f \circ h$ implies $g = h$
- Epimorphism: $f: A \rightarrow B$ epic if $g \circ f = h \circ f$ implies $g = h$
- Limit vs Colimit:
- Limit: Universal cone to diagram
- Colimit: Universal cocone from diagram

Isomorphism of double opposites

- Statement: $(C^{op})^{op} \cong C$
- Proof outline constructs functor $F: C \rightarrow (C^{op})^{op}$ 1. Demonstrate F preserves morphism distinctness (faithful) 2. Show every $(C^{op})^{op}$ morphism originates from C (full) 3. Prove all $(C^{op})^{op}$ objects isomorphic to F image (essentially surjective)
- Conclusion establishes F as category equivalence, confirming isomorphism

1.4 Special objects: initial, terminal, and zero objects

Initial and terminal objects are key concepts in category theory. They provide unique morphisms to or from every object in a category, respectively. These special objects play crucial roles in various mathematical structures.

Zero objects combine properties of both initial and terminal objects. They enable the definition of zero morphisms and are essential in pointed categories. Understanding these concepts is fundamental for grasping more advanced categorical notions.

Special Objects in Categories

Initial and terminal objects

•

Initial object I in category C provides unique morphism from I to every object X in C led to important examples empty set in Set category zero module in R-Mod category and initial algebra in category of algebras over signature

•

Terminal object T in category C receives unique morphism from every object X in C exemplified by singleton set in Set category zero ring in Ring category and terminal coalgebra in category of coalgebras over signature

Uniqueness of categorical extremes

•

Initial objects uniqueness proved through assumed objects I and J with unique morphisms $f: I \rightarrow J$ and $g: J \rightarrow I$ compositions $g \circ f$ and $f \circ g$ yield identity morphisms on I and J respectively demonstrating isomorphism

•

Terminal objects uniqueness established similarly with objects T and S unique morphisms $f: T \rightarrow S$ and $g: S \rightarrow T$ compositions $g \circ f$ and $f \circ g$ result in identity morphisms on T and S proving isomorphism

Zero objects in categories

-

Zero object Z simultaneously initial and terminal in category C provides unique morphisms to and from all objects X in C enables definition of zero morphisms between any two objects

-

Zero objects exemplified by trivial group in Grp category zero vector space in Vect category and zero object in abelian categories

Relationships among categorical objects

-

Zero object special case of both initial and terminal objects not all categories possess initial terminal or zero objects

-

Zero objects existence allows definition of kernels and cokernels in certain categories

-

Duality principle reveals initial object in C becomes terminal object in opposite category C^{op} terminal object in C becomes initial in C^{op} zero object remains zero in C^{op}

-

Categories with zero objects form special class (pointed categories) crucial for defining exact sequences and abelian categories

Unit 2 – Functors and Natural Transformations

2.1 Types of functors: covariant, contravariant, and bifunctors

Functors are the bridges between categories, allowing us to translate structures and relationships. Covariant functors preserve direction, contravariant functors reverse it, and bifunctors take two inputs, each playing a unique role in category theory.

These functor types are crucial for understanding how mathematical structures relate. From preserving group structures to reversing set inclusions and combining vector spaces, functors provide powerful tools for exploring and connecting diverse mathematical concepts.

Understanding Functor Types

Types of functors

- Covariant functors map between categories preserving direction of original morphisms maintain composition and identity morphisms denoted as $F : C \rightarrow D$ (Forgetful functor)
- Contravariant functors reverse direction of morphisms between categories invert composition denoted as $F : C^{op} \rightarrow D$ (Power set functor)
- Bifunctors take two arguments from two categories map to a third category denoted as $F : C \times D \rightarrow E$ (Tensor product)

Examples of functor behavior

- Covariant functor: Forgetful functor from groups to sets preserves structure of morphisms maps group homomorphisms to functions between underlying sets
- Contravariant functor: Power set functor reverses subset inclusions maps larger sets to smaller power sets
- Bifunctor: Tensor product of vector spaces takes two vector spaces produces new vector space with dimension equal to product of input dimensions

Properties of functor types

- Covariant functors preserve direction of morphisms maintain categorical structure respect composition and identities (Homology functor)
- Contravariant functors reverse arrows useful for dual constructions turn colimits into limits (Hom functor)
- Bifunctors generalize binary operations often used to define tensor products provide framework for bilinear maps (Cartesian product)

Applications in category theory

- Covariant functors: 1. Construct free objects 2. Define adjunctions 3. Build universal constructions (Yoneda embedding)

- Contravariant functors: 1. Define duality principles 2. Work with opposite categories 3. Study representable functors (Hom functors)
- Bifunctors: 1. Define bilinear maps 2. Construct product and coproduct functors 3. Study monoidal categories (Tensor product)

2.2 Natural transformations and their properties

Natural transformations are the secret sauce of category theory, connecting functors and revealing hidden structures. They're like bridges between different ways of looking at categories, showing us how seemingly different perspectives can actually be related.

These transformations follow a special rule called the naturality condition, which ensures they play nice with the category's structure. By composing them vertically and horizontally, we can build complex relationships between functors, unlocking powerful insights in mathematics and beyond.

Natural Transformations in Category Theory

Concept of natural transformations

- Maps between functors preserve structure through component morphisms for each object in source category
- Bridge between functors expressing relationships and similarities between different functors
- Key properties include naturality condition and compatibility with functor composition
- Play crucial role in categorical constructions (adjunctions, equivalence of categories, universal properties)

Naturality condition for transformations

- For functors $F, G : C \rightarrow D$ and natural transformation $\alpha : F \rightarrow G$, any morphism $f : X \rightarrow Y$ in C satisfies $G(f) \circ \alpha_X = \alpha_Y \circ F(f)$
- Prove by starting with arbitrary morphism, applying functors and component morphisms, showing equality of resulting compositions
- Ensures preservation of categorical structure and consistency across different objects in category

Composition of natural transformations

- Vertical composition $(\beta \circ \alpha)_X = \beta_X \circ \alpha_X$ for $\alpha : F \rightarrow G$ and $\beta : G \rightarrow H$
- Horizontal composition $(\beta * \alpha)_X = \beta_{G(X)} \circ G(\alpha_X) = K(\alpha_X) \circ \beta_{F(X)}$ for $\alpha : F \rightarrow G$ and $\beta : G \rightarrow K$
- Associativity holds for both vertical $(\gamma \circ \beta) \circ \alpha = \gamma \circ (\beta \circ \alpha)$ and horizontal $(\delta * \gamma) * (\beta * \alpha) = (\delta * \beta) * (\gamma * \alpha)$ compositions
- Interchange law relates vertical and horizontal compositions

Examples of natural isomorphisms

- Natural transformation with each component an isomorphism, inverse natural transformation exists
- Canonical isomorphism between vector space and double dual

- Yoneda lemma isomorphism in category theory
- Currying isomorphism in cartesian closed categories
- Construct by identifying bijective correspondences between functor outputs and proving naturality
- Applications include establishing category equivalence, defining adjunctions, characterizing universal properties

2.3 Functor categories and the Yoneda lemma

Functor categories organize functors and natural transformations, providing a powerful framework for studying relationships between categories. They allow us to view functors as objects and natural transformations as morphisms, creating a rich structure for analysis.

The Yoneda lemma is a cornerstone of category theory, establishing a deep connection between objects and their relationships. It reveals how objects are determined by their interactions within a category, offering new perspectives on categorical structures and their properties.

Functor Categories

Functor categories and components

- Functor category organizes functors between categories C and D as objects and natural transformations as morphisms denoted as $[C, D]$ or D^C
- Objects represent functors $F : C \rightarrow D$ mapping category C to category D (homomorphisms)
- Morphisms consist of natural transformations $\alpha : F \rightarrow G$ between functors F and G (structure-preserving maps)
- Composition follows vertical composition of natural transformations preserving functor properties
- Identity morphism corresponds to identity natural transformation for each functor maintaining structure

Yoneda lemma and interpretations

- Yoneda lemma establishes natural bijection $\text{Nat}(y(c), F) \cong F(c)$ for locally small category C , object $c \in C$, and functor $F : C \rightarrow \text{Set}$
- Interpretation reveals one-to-one correspondence between natural transformations from representable functors to F and elements of $F(c)$
- Objects in category determined by relationships to other objects through morphisms
- Allows studying objects through interactions within category structure
- Provides embedding of any category into category of functors expanding analytical tools

Applications of Yoneda lemma

- Representable functors take form $\text{Hom}(c, -)$ for object c in category
- Yoneda lemma identifies representable functors by showing natural isomorphisms
- Universal elements correspond to natural isomorphisms between functor and representable functor
- Proves properties of adjoint functors using Yoneda lemma's bijection

- Demonstrates representable functors preserve limits enhancing categorical analysis

Yoneda embedding properties

- Yoneda embedding defined as functor $y : C \rightarrow [C^{op}, \text{Set}]$ with $y(c) = \text{Hom}(-, c)$
- Fully faithful proof shows bijective map $\text{Hom}_C(a, b) \rightarrow \text{Hom}_{[C^{op}, \text{Set}]}(y(a), y(b))$ for any $a, b \in C$
- Consequences include embedding C into category of presheaves on C
- Every category viewed as full subcategory of functor category expanding analytical scope
- Yoneda embedding preserves all limits existing in C maintaining structural properties
- Density theorem states every functor $F : C^{op} \rightarrow \text{Set}$ as colimit of representable functors generalizing representation theory

Unit 3 – Limits and Colimits

3.1 Universal properties and representable functors

Universal properties and representable functors are key concepts in category theory. They provide a unified approach to defining and studying mathematical structures across different categories, allowing for abstract characterizations and powerful insights.

The Yoneda lemma, a fundamental result in this area, establishes a deep connection between objects in a category and certain functors. This lemma has far-reaching applications, from proving properties of embeddings to studying natural transformations between functors.

Universal Properties and Representable Functors

Examples of universal properties

- Universal property concept characterizes objects through relationships with other objects defines them up to unique isomorphism
- Set theory examples include terminal object (singleton set) and initial object (empty set)
- Group theory examples feature free group on generator set and quotient group by normal subgroup
- Topology examples encompass product topology and quotient topology
- Importance in category theory provides unified approach to define and study mathematical structures allows abstract characterizations across categories

Concept of representable functors

- Representable functor defined as $\text{Hom}(A, -)$ for object A in category C maps objects to sets of morphisms from A
- Properties include limit preservation and isomorphism reflection
- Connection to universal properties often expressed through representable functors establishes bijection between natural transformations and elements of representing object
- Yoneda lemma establishes fundamental relationship between representable functors and other functors

Construction of Yoneda embedding

- Yoneda embedding constructed as functor $Y : C \rightarrow [C^{op}, \text{Set}]$ maps object A to representable functor $\text{Hom}(-, A)$ and morphism $f : A \rightarrow B$ to natural transformation $\text{Hom}(-, f)$
- Properties include full faithfulness limit and colimit preservation
- Significance allows study of category C through functors $C^{op} \rightarrow \text{Set}$ embeds C into functor and natural transformation category provides concrete representation of abstract categorical concepts

Proof and applications of Yoneda lemma

- Yoneda lemma states natural bijection between natural transformations $\text{Hom}(A, -) \rightarrow F$ and elements of $F(A)$

- Proof outline: 1. Construct bijection explicitly 2. Show naturality of bijection 3. Verify bijectivity
- Applications include characterizing representable functors proving full faithfulness of Yoneda embedding studying natural transformations between functors
- Consequences demonstrate objects in category determined by morphisms into them establish equivalence between small categories and certain functor categories lay foundation for enriched category theory and higher category theory

3.2 Limits, colimits, and their basic properties

Limits and colimits are fundamental concepts in category theory, capturing universal properties of constructions like products and coproducts. They provide a unified framework for understanding various mathematical structures and relationships between objects in a category.

These concepts are essential for abstracting common patterns across different mathematical fields. By focusing on universal properties, limits and colimits allow us to reason about structures without relying on specific constructions, enabling powerful generalizations in category theory.

Fundamental Concepts of Limits and Colimits

Definition of limits and colimits

- Limits
- Terminal object in category of cones over diagram captures universal property
- Product embodies limit of discrete diagram (cartesian product of sets)
- Pullback represents limit of span diagram (fiber product in algebraic geometry)
-

Equalizer manifests limit of parallel pair of morphisms (subset where functions agree)

-

Colimits

- Initial object in category of cocones under diagram encapsulates dual notion
- Coproduct exemplifies colimit of discrete diagram (disjoint union of sets)
- Pushout illustrates colimit of cospan diagram (quotient by generated equivalence relation)
-

Coequalizer demonstrates colimit of parallel pair of morphisms (quotient identifying elements)

-

Diagrams

- Functor from index category to category of interest defines shape
- Shape determines limit or colimit type (discrete, span, cospan, parallel pair)

Universal properties of limits and colimits

- Limits
- Unique morphism exists from any cone to limit cone ensuring universality
-

Resulting diagram commutes preserving structure

-

Colimits

- Unique morphism exists from colimit cocone to any cocone guaranteeing universality
-

Resulting diagram commutes maintaining coherence

-

Importance

- Characterize limits and colimits up to isomorphism enabling abstract manipulation
- Allow reasoning without specific constructions facilitating generalization

Advanced Properties and Applications

Uniqueness of limits and colimits

-

Limits 1. Assume two limit objects L_1 and L_2 with respective cones 2. Construct unique morphisms $f: L_1 \rightarrow L_2$ and $g: L_2 \rightarrow L_1$ using universal properties 3. Prove $g \circ f = id_{L_1}$ and $f \circ g = id_{L_2}$ establishing isomorphism

-

Colimits

- Analogous proof structure leveraging universal properties of colimits
-

Construct isomorphisms between candidate colimit objects

-

Significance

- Ensures unambiguous definition across different constructions
- Guarantees consistency in categorical framework

Preservation of limits by functors

- Limit preservation
- Functor F preserves limits if $F(\lim D) \cong \lim(F \circ D)$ holds

-

Representable functors and right adjoints exemplify limit-preserving functors

-

Limit reflection

- Functor F reflects limits if $\lim(F \circ D) \cong F(L)$ implies $L \cong \lim D$

-

Stronger condition than preservation requiring additional structure

-

Colimit preservation

- Functor F preserves colimits if $F(\operatorname{colim} D) \cong \operatorname{colim}(F \circ D)$ holds

-

Representable functors and left adjoints illustrate colimit-preserving functors

-

Colimit reflection

- Functor F reflects colimits if $\operatorname{colim}(F \circ D) \cong F(C)$ implies $C \cong \operatorname{colim} D$

-

Analogous to limit reflection with dual properties

-

Categorical significance

- Enables transfer of properties between categories (algebraic structures)
- Crucial in studying adjoint functors and Kan extensions (fundamental constructions)

3.3 Special cases: products, coproducts, equalizers, and coequalizers

Limits and colimits are fundamental concepts in category theory, shaping how objects and morphisms interact. This section dives into special cases like products, coequalizers, pullbacks, and pushouts, exploring their construction and universal properties.

Understanding these special cases is crucial for grasping the broader concept of limits and colimits. We'll examine how they're computed in different categories and use commutative diagrams to verify their properties.

Special Cases of Limits and Colimits

Products and coequalizers

- Products
- Limit of diagram consisting of objects without morphisms between them terminates in category of cones over diagram

- Cartesian product of sets exemplifies in Set category

-

Product topology demonstrates in Top category

-

Coproducts

- Colimit of diagram consisting of objects without morphisms between them initiates in category of cocones under diagram
- Disjoint union of sets illustrates in Set category

-

Free product of groups showcases in Grp category

-

Equalizers

- Limit of parallel morphism pair terminates in category of equalizing cones
- Subset of elements where two functions agree represents in Set category

-

Kernel of difference between two group homomorphisms exemplifies in Grp category

-

Coequalizers

- Colimit of parallel morphism pair initiates in category of coequalizing cocones
- Quotient set by equivalence relation generated by function pair demonstrates in Set category
- Quotient ring by ideal generated by image differences illustrates in Rng category

Construction of pullbacks and pushouts

- Pullbacks
- Limit of cospan diagram constructs using products and equalizers:
 1. Form product of two objects at cospan base
 2. Take equalizer of two compositions from product to apex

-

Fiber product in algebraic geometry serves as example

-

Pushouts

- Colimit of span diagram constructs using coproducts and coequalizers:
 1. Form coproduct of two objects at span base

2. Take coequalizer of two compositions from apex to coproduct

- Amalgamated free product in group theory demonstrates concept

Universal properties of limits

- Universal property of pullbacks
- Terminates in category of cones over cospan diagram
- Uniquely factorizes through any other cone
-

Preserves monomorphisms (injective functions)

-

Universal property of pushouts

- Initiates in category of cocones under span diagram
- Uniquely factorizes from any other cocone
- Preserves epimorphisms (surjective functions)

Computation of categorical limits

- Set category exercises
- Products compute as Cartesian products (ordered pairs)
- Coproducts calculate as disjoint unions (tagged unions)
- Equalizers determine as subsets (common elements)
-

Coequalizers find as quotient sets (equivalence classes)

-

Group category exercises

- Products construct as direct products (element-wise operation)
- Coproducts compute as free products (words with alternating elements)
- Equalizers calculate as kernels (elements mapping to identity)
-

Coequalizers determine as quotient groups (cosets)

-

Topological space category exercises

- Products compute with product topology (box topology)
- Coproducts calculate with disjoint union topology (pasting topology)

- Equalizers determine as subspaces (induced topology)

-

Coequalizers find as quotient spaces (identification topology)

-

Commutative diagram techniques

- Commutativity of pullback and pushout squares verify through path equality
- Universal properties apply to prove uniqueness of constructions
- Functoriality of limits and colimits utilize for preservation under functors

3.4 Completeness and cocompleteness of categories

Categories can be complete, containing all small limits, or cocomplete, containing all small colimits. These properties are crucial for understanding universal constructions and preserving structure between categories.

Set theory exemplifies completeness and cocompleteness through products, equalizers, coproducts, and coequalizers. This pattern extends to other common categories like groups, rings, and topological spaces, forming a foundation for categorical analysis.

Completeness and Cocompleteness in Categories

Definition of complete categories

- Complete categories possess all small limits
- Small limit results from functor mapping small category to given category
- Cocomplete categories contain all small colimits
- Small colimit stems from functor connecting small category to given category
- Limits act as universal cones exemplified by products, equalizers, pullbacks
- Colimits function as universal cocones demonstrated through coproducts, coequalizers, pushouts
- Small category comprises set of objects rather than proper class

Completeness of set category

- Set category exhibits completeness through existence of products for any set family
- Equalizers exist for all parallel function pairs
- Small limits constructed from products and equalizers
- Set category demonstrates cocompleteness via coproducts (disjoint union) for any set family
- Coequalizers exist for all parallel function pairs
- Small colimits built from coproducts and coequalizers
- Proof strategy involves showing existence of products, equalizers, coproducts, coequalizers

- Utilizes fact that all limits and colimits derivable from these constructions

Completeness across common categories

- Group category complete with products and equalizers cocomplete with free products and coequalizers
- Ring category complete featuring products and equalizers cocomplete with tensor products and coequalizers
- Topological spaces category complete with products and equalizers cocomplete through disjoint unions and coequalizers
- Vector spaces over field both complete and cocomplete with componentwise computation of limits and colimits
- Partially ordered sets as categories exhibit completeness when all subsets have least upper bounds
- Partially ordered sets demonstrate cocompleteness when all subsets possess greatest lower bounds

Significance of categorical completeness

- Structure preservation achieved through limit-preserving functors between complete categories
- Adjoint functor theorem relates to completeness and cocompleteness
- New category construction inherits completeness and cocompleteness from target or base categories (functor categories, slice categories)
- Universal constructions embodied by limits and colimits define crucial categorical concepts
- Algebraic theories yield complete and cocomplete model categories vital in universal algebra and categorical logic
- Elementary toposes in topos theory exhibit completeness and cocompleteness underpinning categorical foundations of mathematics
- Homotopy theory leverages interplay between completeness, cocompleteness, and homotopy in model categories
- Completeness and cocompleteness serve as fundamental properties for modeling mathematical structures in category theory as a foundation

Unit 4 – Adjoint Functors

4.1 Definition and properties of adjoint functors

Adjoint functors are a powerful tool in category theory, allowing bidirectional transformations between categories. They consist of a left adjoint F and a right adjoint G , connected by a natural bijection between morphism sets.

The uniqueness theorem ensures that adjoint functors are essentially unique up to natural isomorphism. This property, along with the adjoint functor theorem and limit preservation, makes adjoints crucial for understanding categorical structures and constructing new functors.

Adjoint Functors: Definition and Fundamental Properties

Adjoint functors between categories

- Adjoint functors form a powerful pair of functors between two categories enabling bidirectional transformations
- Left adjoint $F : C \rightarrow D$ maps objects and morphisms from category C to D
- Right adjoint $G : D \rightarrow C$ maps objects and morphisms from category D to C
- Natural bijection establishes a correspondence between morphisms in both categories
- $\text{Hom}_D(F(A), B) \cong \text{Hom}_C(A, G(B))$ for all objects A in C and B in D
- Enables translation of morphisms between categories (homomorphisms, continuous functions)
- Unit and counit serve as universal arrows in the adjunction
- Unit $\eta : 1_C \rightarrow GF$ compares identity functor on C with composition GF
- Counit $\varepsilon : FG \rightarrow 1_D$ compares composition FG with identity functor on D
- Triangle identities ensure coherence of the adjunction
- $G\varepsilon \circ \eta G = 1_G$ verifies compatibility of unit and counit with G
- $\varepsilon F \circ F\eta = 1_F$ verifies compatibility of unit and counit with F

Uniqueness of adjoint functors

- Uniqueness theorem guarantees essential uniqueness of adjoint functors
- If $F \dashv G$ and $F \dashv G'$, then $G \cong G'$ natural isomorphism between right adjoints
- If $F \dashv G$ and $F' \dashv G$, then $F \cong F'$ natural isomorphism between left adjoints
- Proof strategy involves constructing natural isomorphisms
- Utilize universal property of adjunctions to define component morphisms
- Verify naturality and isomorphism conditions
- Yoneda lemma plays crucial role in establishing uniqueness
- Relates natural transformations to representable functors

- Allows reduction of functor equality to object-wise equality
- Natural transformations connect functors in the uniqueness proof
- Between functors G and G' for right adjoint uniqueness
- Between functors F and F' for left adjoint uniqueness

Applications and Implications of Adjoint Functors

Adjoint functor theorem

- General adjoint functor theorem provides conditions for existence of left adjoints
- Solution set condition ensures "smallness" of potential candidates
- Applicable to wide range of categories (abelian groups, topological spaces)
- Special adjoint functor theorem simplifies conditions for locally small, complete categories
- Preservation of small limits replaces solution set condition
- Useful in algebraic categories (rings, modules)
- Freyd's adjoint functor theorem applies to locally presentable categories
- Characterizes existence of left adjoints through accessibility conditions
- Relevant in categorical logic and topos theory
- Implications extend to various areas of mathematics
- Constructing new functors by composing known adjoints
- Characterizing categorical properties (limits, colimits, exponentials)

Limit preservation by adjoints

- Right adjoints preserve limits allowing transfer of limit structures
- Proof utilizes natural isomorphism of hom-sets
- Applies to products, pullbacks, equalizers
- Left adjoints preserve colimits enabling preservation of colimit structures
- Dual statement to right adjoints
- Applies to coproducts, pushouts, coequalizers
- Consequences impact various mathematical constructions
- Forgetful functors often have left adjoints (free constructions)
- Free constructions as left adjoints preserve colimits
- Examples illustrate practical applications
- Free group functor preserves coproducts (disjoint unions)
- Forgetful functor from groups to sets preserves products (Cartesian products)

- Relationship to representable functors connects to broader category theory
- Adjoint functors naturally isomorphic to representable functors
- Enables application of Yoneda lemma and related results

4.2 Unit and counit of adjunction

Adjunctions are powerful tools in category theory, linking functors between categories. The unit and counit of an adjunction provide a way to measure how close these categories are to being retracts of each other.

Understanding units and counits is crucial for grasping adjunctions. These natural transformations, along with triangle identities, form the foundation for constructing and analyzing adjunctions, helping us explore relationships between different mathematical structures.

Unit and Counit of Adjunction

Unit and counit of adjunctions

- Unit of adjunction
- Natural transformation $\eta : 1_C \rightarrow GF$ maps identity functor on C to composition of F and G
- Maps objects in category C to objects in C via GF creates a "round trip" through D
-

For each object X in C, $\eta_X : X \rightarrow GF(X)$ represents the unit's action on X

•

Counit of adjunction

- Natural transformation $\varepsilon : FG \rightarrow 1_D$ maps composition of F and G to identity functor on D
- Maps objects in category D to objects in D via FG creates a "round trip" through C
-

For each object Y in D, $\varepsilon_Y : FG(Y) \rightarrow Y$ represents the counit's action on Y

•

Naturality of unit

- For any morphism $f : X \rightarrow X'$ in C, the following diagram commutes:
$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & GF(X) \\ \downarrow f & & \downarrow GF(f) \\ X' & \xrightarrow{\eta_{X'}} & GF(X') \end{array}$$
-

Ensures consistency of unit with morphisms in C (preserves structure)

•

Naturality of counit

- For any morphism $g : Y \rightarrow Y'$ in D, the following diagram commutes:
$$\begin{array}{ccc} FG(Y) & \xrightarrow{FG(g)} & FG(Y') \\ \downarrow \varepsilon_Y & & \downarrow \varepsilon_{Y'} \\ Y & \xrightarrow{g} & Y' \end{array}$$

- Ensures consistency of counit with morphisms in D (preserves structure)

Triangle identities for adjunctions

- Triangle identity for F
- Composition $F \xrightarrow{F\eta} FGF \xrightarrow{\varepsilon F} F$ equals identity on F
-

Ensures F "cancels out" the round trip through G (Set, Group)

•

Triangle identity for G

- Composition $G \xrightarrow{\eta G} GFG \xrightarrow{G\varepsilon} G$ equals identity on G
-

Ensures G "cancels out" the round trip through F (Vector spaces, Modules)

•

Proof strategy 1. Use naturality of unit and counit to manipulate diagrams 2. Apply definitions of functor composition to simplify expressions 3. Utilize properties of identity morphisms to complete the proof 4. Consider specific examples (Free group functor, Forgetful functor)

Correspondence of adjunctions vs natural transformations

- Adjunction as a quadruple $(F, G, \eta, \varepsilon)$ defines relationship between categories
- $F: C \rightarrow D$ (left adjoint) maps objects and morphisms from C to D
- $G: D \rightarrow C$ (right adjoint) maps objects and morphisms from D to C
- $\eta: 1_C \rightarrow GF$ (unit) measures how far C is from being a retract of D
-

$\varepsilon: FG \rightarrow 1_D$ (counit) measures how far D is from being a retract of C

•

Bijjective correspondence establishes equivalence between adjunctions and natural transformations

- Given an adjunction, unit and counit automatically satisfy triangle identities
-

Given unit and counit satisfying triangle identities, construct unique adjunction (Free-forgetful, Product-diagonal)

•

Hom-set isomorphism $\text{Hom}_D(F(X), Y) \cong \text{Hom}_C(X, G(Y))$ relates morphisms in C and D

- Bijection between morphisms $F(X) \rightarrow Y$ in D and morphisms $X \rightarrow G(Y)$ in C
-

Relationship to unit and counit through natural transformations

-

Uniqueness of adjunction demonstrates fundamental nature of the concept

- Prove that any two adjunctions with the same unit (or counit) are isomorphic
- Implies adjunctions are determined by their unit or counit (up to isomorphism)

Construction of adjunctions from units

-

Construction from unit 1. Given $\eta : 1_C \rightarrow GF$, define left adjoint F using universal property 2. Construct counit ε using properties of G and η 3. Verify triangle identities to ensure adjunction properties

-

Construction from counit 1. Given $\varepsilon : FG \rightarrow 1_D$, define right adjoint G using universal property 2. Construct unit η using properties of F and ε 3. Verify triangle identities to ensure adjunction properties

-

Universal property approach leverages categorical definitions

- Use unit to define universal arrows from X to G (initial objects)

-

Use counit to define universal arrows from F to Y (terminal objects)

-

Examples of adjunction construction demonstrate practical applications

- Free-forgetful adjunction (groups and sets)
- Product-diagonal adjunction (cartesian product and diagonal functor)
- Exponential-product adjunction (function spaces and cartesian product)

4.3 Examples and applications of adjunctions

Adjoint functors are powerful tools in category theory, connecting different mathematical structures. They pair up functors that work together, like free and forgetful functors in algebra, or product and exponential functors in cartesian closed categories.

These functors solve universal problems, construct important objects, and reveal deep connections. From free groups to tensor products, adjunctions pop up everywhere, linking seemingly disparate areas of math and providing a unified framework for understanding categorical structures.

Adjoint Functors and Their Applications

Adjoint functors in categories

- Free-forgetful adjunction pairs functors constructing free objects and stripping away structure
- Free functor builds objects with minimal structure (free groups, free modules)
- Forgetful functor removes algebraic structure retaining underlying set

- Group-Set adjunction connects groups and sets
- Free functor generates free group from set elements as generators
- Forgetful functor views group solely as collection of elements
- Vector space-Set adjunction relates vector spaces to sets
- Free functor creates vector space with set as basis
- Forgetful functor considers only underlying set of vectors
- Abelianization functor maps groups to abelian groups preserving commutativity
- Tensor-Hom adjunction in module categories connects tensor products and homomorphisms

Adjunctions for universal mapping

- Universal property of adjunctions establishes natural bijection between morphism sets
- Adjoint functors solve universal problems by constructing objects with specific properties
- Initial objects serve as sources for unique morphisms to all objects
- Terminal objects receive unique morphisms from all objects
- Products combine objects while coproducts offer choices between objects
- Free group construction universally embeds set into group category
- Tensor product universally linearizes bilinear maps between modules
- Adjoint functor theorem provides conditions for existence of adjoints using solution sets

Exponential objects and adjunctions

- Cartesian closed categories possess internal hom-objects and terminal objects
- Exponential objects act as right adjoints to product functors
- $(-) \times Y \dashv (-)^Y$ formalizes currying in category theory
- Curry-Howard-Lambek correspondence connects logic, computation, and category theory
- Types correspond to objects, functions to morphisms, propositions to types
- Product-exponential adjunction establishes isomorphism between hom-sets
- $\text{Hom}(X \times Y, Z) \cong \text{Hom}(X, Z^Y)$ enables currying and function evaluation
- Power set functor right adjoints to singleton functor mapping elements to one-element sets

Adjunctions in categorical constructions

- Limits and colimits universally construct objects from diagrams in categories
- Adjoint functor pairs for limits and colimits connect constant and (co)limit functors
- $\Delta \dashv \lim$ relates constant diagrams to limit objects
- $\text{colim} \dashv \Delta$ connects colimit objects to constant diagrams

- Kan extensions generalize limits and colimits as adjoint functors
- Left Kan extensions act as best approximations to left adjoints
- Right Kan extensions approximate right adjoints
- Monads arise from adjunctions describing algebraic structures
- Eilenberg-Moore construction yields category of algebras for a monad
- Kleisli construction produces category of free algebras for a monad
- Galois connections form order-theoretic adjunctions between partially ordered sets
- Adjoint functors in algebraic theories connect free algebras to underlying sets
- Free algebra functor generates algebras from sets of generators
- Forgetful functor views algebras as sets forgetting operations

Unit 5 – Cartesian Closed Categories

5.1 Definition and properties of cartesian closed categories

Cartesian closed categories (CCCs) are a powerful concept in category theory. They generalize function spaces, allowing us to work with morphisms between objects as if they were objects themselves. This abstraction is crucial for modeling computation and logic.

CCCs have special properties like finite products and exponential objects. These features enable us to manipulate functions in flexible ways, similar to currying in programming. Understanding CCCs helps us grasp the deep connections between category theory and functional programming.

Cartesian Closed Categories: Definition and Properties

Definition of cartesian closed categories

- Cartesian closed category (CCC) exhibits finite products, exponential objects, and natural isomorphism between hom-sets and exponential objects
- Key properties encompass closure under products and exponentials, terminal object existence, binary products, and exponential objects for any pair of objects
- Adjunction property manifests as $\text{Hom}(A \times B, C) \cong \text{Hom}(A, C^B)$ facilitating function manipulation
- Currying in CCCs establishes correspondence between multi-argument and single-argument functions enabling flexible function representation

Cartesian closed categories vs exponential objects

- Exponential objects function as internal hom-objects representing morphisms from B to C as C^B
- Evaluation morphism $\text{ev} : C^B \times B \rightarrow C$ applies functions to arguments within the category
- Exponential transpose yields unique $\tilde{f} : A \rightarrow C^B$ for any $f : A \times B \rightarrow C$ enabling function transformation
- CCCs provide categorical framework for lambda calculus supporting functional programming concepts

Proof of cartesian closure

1. Verify terminal object existence ensuring universal property satisfaction
2. Confirm binary products presence with projection morphisms and universal property
3. Demonstrate exponential objects for all object pairs with evaluation morphism
4. Establish natural isomorphism $\text{Hom}(A \times B, C) \cong \text{Hom}(A, C^B)$ using universal property
5. Verify naturality in A , B , and C through commutative diagrams - Conclusion: Category $\mathcal{C}$ satisfies cartesian closure conditions

Generalization of function spaces

- CCC generalizes set theory function spaces extending B^A concept to exponential object C^B
- Preserves evaluation $(f, x) \mapsto f(x)$ and currying $f : A \times B \rightarrow C$ to $\tilde{f} : A \rightarrow C^B$

- Examples beyond Set include topological spaces and vector spaces categories
- Internal logic of CCCs provides semantic framework for typed lambda calculi supporting programming language theory

5.2 Exponential objects and evaluation morphisms

Exponential objects in category theory generalize function spaces, denoted as B^A . They have a universal property that ensures a unique morphism for any object X and morphism $f: X \times A \rightarrow B$, establishing a bijective correspondence between certain Hom-sets.

Cartesian closed categories, like Set and Cat, have all finite products and exponential objects. The evaluation morphism $ev: B^A \times A \rightarrow B$ represents function application. Exponential objects are unique up to isomorphism and connect to adjunctions between product and exponential functors.

Exponential Objects in Category Theory

Definition of exponential objects

- Exponential objects generalize function spaces in category theory denoted as B^A for objects A and B in a category C (Set, Top)
- Universal property ensures existence of unique morphism for any object X and morphism $f: X \times A \rightarrow B$ establishing bijective correspondence between $\text{Hom}(X \times A, B)$ and $\text{Hom}(X, B^A)$
- Cartesian closed categories possess all finite products and exponential objects (Cat, Set)
- Currying transforms function of multiple arguments into sequence of functions ($f(x, y) \rightarrow f(x)(y)$)

Construction of evaluation morphisms

- Evaluation morphism $ev: B^A \times A \rightarrow B$ represents function application in the category
- Construction steps: 1. Start with identity morphism $id: B^A \rightarrow B^A$ 2. Apply universal property to obtain $ev: B^A \times A \rightarrow B$
- Commutative diagram illustrates relationship between ev and other morphisms

Uniqueness of exponential objects

- Uniqueness theorem states if B^A and C both satisfy universal property, then $B^A \cong C$
- Proof outline: 1. Assume two objects satisfying universal property 2. Construct isomorphisms between them using universal property 3. Show composition of isomorphisms is identity
- Yoneda lemma relates to uniqueness of representable functors

Connection to adjunctions

- Adjunction pairs functors $F: C \rightarrow D$ and $G: D \rightarrow C$ with natural bijection
- Exponential-product adjunction:
- Left adjoint: product functor $(- \times A)$
- Right adjoint: exponential functor $(B^{_})$
- Natural isomorphism $\text{Hom}(X \times A, B) \cong \text{Hom}(X, B^A)$ establishes correspondence

- Curry-uncurry operations correspond to unit and counit of adjunction
- Cartesian closed categories characterized by existence of this adjunction for all objects

5.3 Examples of cartesian closed categories

Cartesian closed categories are a fundamental concept in category theory, combining terminal objects, binary products, and exponential objects. They provide a powerful framework for understanding function spaces and currying in various mathematical contexts.

Set theory and category theory both showcase cartesian closed structures. The category of sets (Set) and small categories (Cat) demonstrate these properties, with unique features like function spaces and functor categories serving as exponential objects.

Cartesian Closed Categories in Set Theory and Category Theory

Cartesian closure of sets

- Definition of a cartesian closed category encompasses existence of terminal object representing final object in category allowing unique morphism from any object existence of binary products allowing pairing of objects and existence of exponential objects enabling function objects
- Category of sets (Set) exemplifies cartesian closed structure with singleton set as terminal object allowing unique function from any set cartesian product of sets as binary products and function space B^A for sets A and B as exponential objects
- Exponential objects in Set comprise function space B^A for sets A and B establishing bijection between $Hom(A \times B, C)$ and $Hom(A, C^B)$
- Natural isomorphism for exponential objects formulated as $Hom(A \times B, C) \cong Hom(A, C^B)$ demonstrates correspondence between functions of multiple arguments and curried functions
- Verification of cartesian closed axioms for Set involves proving existence of terminal object binary products and exponential objects along with natural isomorphisms

Cartesian closure of small categories

- Category of small categories (Cat) exhibits cartesian closed structure with terminal object as category containing one object and one morphism
- Binary products in Cat constructed as product of categories with objects as pairs from constituent categories and morphisms as pairs of morphisms from constituent categories
- Exponential objects in Cat manifest as functor category C^D for small categories C and D with objects as functors from D to C and morphisms as natural transformations between functors
- Verification of cartesian closed axioms for Cat involves proving existence of terminal object binary products and exponential objects along with natural isomorphisms
- Natural isomorphism for exponential objects in Cat demonstrates correspondence between functors and natural transformations in product and exponential categories

Examples of cartesian closed categories

- Category of topological spaces (Top) with objects as topological spaces and morphisms as continuous functions

- Cartesian closed subcategories of Top include category of compactly generated Hausdorff spaces and category of locally compact Hausdorff spaces
- Category of vector spaces over a field exhibits cartesian closed structure (finite-dimensional vector spaces)
- Category of modules over a commutative ring demonstrates cartesian closed properties (finitely generated projective modules)
- Cartesian closed categories in computer science encompass category of domains (continuous lattices) and category of complete partial orders (CPOs)
- Topos theory connects to cartesian closed categories with elementary toposes as cartesian closed categories possessing subobject classifier

Properties of exponential objects

- General properties of exponential objects include adjunction between product and exponential functors and currying/uncurrying operations
- Exponential objects in Set manifest as set of functions between two sets with evaluation and co-evaluation maps
- Exponential objects in Cat take form of functor categories with natural transformations as morphisms
- Exponential objects in topological spaces utilize compact-open topology on function spaces ensuring continuity of evaluation map
- Universal property of exponential objects guarantees uniqueness up to isomorphism and preservation under equivalence of categories
- Exponential preserving functors maintain structure of exponential objects between categories relating to cartesian closed functors
- Applications of exponential objects appear in higher-order functions in programming languages and internal language of cartesian closed categories

Unit 6 – Subobject Classifiers and Topoi

6.1 Subobjects and characteristic functions

Subobjects in categories extend the idea of "parts" or "subsets" to abstract mathematical structures. They're represented by equivalence classes of monomorphisms, allowing us to compare and analyze substructures across different mathematical domains.

Characteristic functions of subobjects generalize indicator functions, connecting subobjects to logic and type theory. This concept finds applications in various fields, from set theory to algebraic geometry and computer science, highlighting its versatility and importance.

Subobjects in Categories

Definition of subobjects

- Subobject represents equivalence class of monomorphisms capturing notion of "part" or "subset" in abstract categories
- Two monomorphisms deemed equivalent if they factor through each other allowing different representations of same substructure
- Denoted $m : A \rightarrow B$, where m is a monomorphism embedding substructure A into larger structure B
- Generalizes familiar concepts like subsets (Set), subgroups (Group), and subspaces (Top)

Subobjects vs monomorphisms

- Monomorphisms serve as representatives for subobjects but not in one-to-one correspondence
- Factorization property defines equivalence: $f : A \rightarrow C$ and $g : B \rightarrow C$ represent same subobject if isomorphisms $h : A \rightarrow B$ and $k : B \rightarrow A$ exist with $f = g \circ h$ and $g = f \circ k$
- Subobjects form partially ordered set based on factorization relation enabling comparison of "size" or "inclusion"
- Distinction crucial for understanding abstract structure preservation in category theory

Characteristic functions of subobjects

- Morphism $\chi_m : B \rightarrow \Omega$ where Ω is subobject classifier generalizing indicator functions
- Constructed using universal property of subobject classifier ensuring unique factorization through true morphism $true : 1 \rightarrow \Omega$
- Exhibits uniqueness for each subobject and pullback property allowing recovery of subobject
- Connects to internal logic and type theory in elementary toposes

Applications in various categories

- Set: Subobjects as subsets with characteristic functions acting as indicator functions (0 and 1)
- Top: Open subsets as subobjects utilizing local homeomorphisms for characteristic functions
- Algebraic geometry: Closed subschemes as subobjects related to ideal sheaves

- Computer science: Modeling data structures (trees, graphs) and type systems in programming languages
- Logic: Internal logic of toposes leveraging subobject classifiers for intuitionistic reasoning

6.2 Definition and properties of subobject classifiers

Subobject classifiers are powerful tools in category theory that generalize truth values. They're objects with special properties that allow us to classify subobjects in a category, providing a way to think about "truth" in different mathematical contexts.

In $\mathbf{Set}$, the subobject classifier is the two-element set $\{0,1\}$, while in $\mathbf{Top}$, it's the Sierpinski space. These examples show how subobject classifiers adapt to different mathematical structures, enabling us to reason about truth and logic in various settings.

Subobject Classifiers in Category Theory

Definition of subobject classifiers

- Subobject classifier denotes object Ω in category $\mathcal{C}$ with terminal object 1 and morphism $true : 1 \rightarrow \Omega$ (truth arrow) generalizing concept of truth values
- Universal property characterizes subobject classifiers ensuring for every monomorphism $m : A \rightarrow B$ in $\mathcal{C}$, unique characteristic function $\chi_m : B \rightarrow \Omega$ exists making following diagram a pullback:

$$\begin{array}{ccc} A & \xrightarrow{m} & B \\ \downarrow \chi_m & \lrcorner & \downarrow true \\ 1 & \xrightarrow{} & \Omega \end{array}$$
- Pullback property establishes A as pullback of χ_m along $true$, capturing notion of subobject
- Bijective correspondence links subobjects of B with morphisms $B \rightarrow \Omega$, providing classification of subobjects

Uniqueness of subobject classifiers

- Assume two subobject classifiers $(\Omega, true)$ and $(\Omega', true')$ in category
- Construct isomorphism $f : \Omega \rightarrow \Omega'$ using universal property of Ω' , defining f as characteristic function of $true : 1 \rightarrow \Omega$
- Construct inverse $g : \Omega' \rightarrow \Omega$ using universal property of Ω , defining g as characteristic function of $true' : 1 \rightarrow \Omega'$
- Prove $f \circ g = id_{\Omega'}$ and $g \circ f = id_{\Omega}$ using uniqueness of characteristic functions
- Conclude Ω and Ω' are isomorphic, establishing uniqueness up to isomorphism

Existence in Set and Top

- Subobject classifier in $\mathbf{Set}$: 1. Define $\Omega = \{0, 1\}$ (two-element set) 2. Establish $true : 1 \rightarrow \{0, 1\}$ mapping singleton to 1 3. Characteristic functions correspond to indicator functions of subsets
- Subobject classifier in $\mathbf{Top}$ (category of topological spaces): 1. Define $\Omega = \{0, 1\}$ with Sierpinski topology (open sets: $\{\emptyset, \{1\}, \{0, 1\}\}$) 2. Establish $true : 1 \rightarrow \Omega$ mapping point to 1 3. Characteristic functions correspond to open subsets of spaces
- Non-existence in some categories demonstrated by category of groups lacking subobject classifier

Relationship to truth values

- Topos represents category with additional structure: cartesian closed, possesses all finite limits and colimits, has subobject classifier
- Subobject classifier generalizes truth values with elements of Ω representing truth values and morphisms $1 \rightarrow \Omega$ corresponding to propositions
- Internal logic of topos enabled by subobject classifier allowing logical operations (conjunction, disjunction, negation, implication) with Heyting algebra structure on Ω
- Relationship to intuitionistic logic highlighted by law of excluded middle potentially not holding, different toposes yielding different logical systems
- Non-classical truth values exemplified by sheaf toposes (truth values as open sets) and effective topos (realizability interpretation of logic)

6.3 Elementary topoi and their basic properties

Elementary topoi are powerful mathematical structures that combine category theory and logic. They provide a framework for studying sets, functions, and logical operations in a unified way, offering insights into foundations of mathematics and computer science.

In this section, we'll explore the definition and examples of elementary topoi, including Set, sheaves, and presheaves. We'll also delve into the crucial concept of subobject classifiers and their role in topos logic and intuitionistic reasoning.

Foundations of Elementary Topoi

Definition and examples of elementary topos

- Elementary topos category with finite limits supports exponentials and subobject classifier enables powerful categorical logic
- Set category encompasses sets and functions fundamental to mathematics provides concrete example of topos structure
- $\text{Sh}(X)$ category of sheaves on topological space X generalizes local-to-global properties crucial in algebraic geometry
- G-Set category of G -sets for group G represents group actions on sets illustrates algebraic aspects of topoi
- Presheaf categories functors from small category to Set model varying data over index category
- Effective topos incorporates realizability theory bridges constructive mathematics and computer science

Proof of subobject classifier existence

- Subobject classifier Ω object in topos with true morphism $1 \rightarrow \Omega$ characterizes subobjects
- Proof outline constructs Ω builds true morphism establishes universal property: 1. Construct Ω as object of subobjects of 1 (terminal object) 2. Define true: $1 \rightarrow \Omega$ representing maximal subobject 3. For monomorphism $m: A \rightarrow B$ show unique $\chi: B \rightarrow \Omega$ exists 4. Demonstrate m as pullback of true along χ completing universal property

Advanced Properties and Connections

Role of subobject classifier in topos logic

- Internal logic language reasons about objects and morphisms within topos framework
- Subobject classifier represents truth values enables logical operations (conjunction disjunction negation implication)
- Facilitates predicate and quantifier formulation supports power object construction
- Heyting algebra structure on subobject classifier models intuitionistic logic

Cartesian closed categories in elementary topoi

- Cartesian closed category (CCC) has finite products and exponentials for all object pairs
- CCCs in topoi enable internal function spaces support higher-order logic facilitate curry-uncurry isomorphism
- Natural number object definition supported by CCC structure
- Provides semantic model for typed lambda calculi connecting category theory and computer science

Elementary topoi and intuitionistic logic

- Intuitionistic logic emphasizes constructive reasoning without assuming law of excluded middle
- Subobject classifier models intuitionistic truth values internal logic inherently intuitionistic
- Topoi serve as models for intuitionistic set theory (IZF)
- Kripke-Joyal semantics interprets intuitionistic logic in topoi
- Connects to constructive mathematics and proof theory advancing foundations of mathematics

Unit 7 – Elementary Topoi and Examples

7.1 Axioms and properties of elementary topoi

Elementary topoi generalize properties of the category of sets, requiring finite limits, colimits, and cartesian closure. They feature a subobject classifier Ω , which acts as a generalized truth values object, allowing for internal logic within the topos.

The axioms of elementary topoi provide a rich structure for studying categorical logic and set-like objects. The subobject classifier plays a crucial role, enabling the representation of membership, subobjects, and power objects, while supporting an intuitionistic internal logic.

Foundations of Elementary Topoi

Axioms of elementary topos

- Elementary topos generalizes properties of category of sets as category with additional structure (Grothendieck topos)
- Axioms require all finite limits and colimits exist including products, equalizers, coproducts, coequalizers (pullbacks, pushouts)
- Cartesian closure ensures internal hom objects and exponential objects exist for any pair of objects (B^A)
- Subobject classifier Ω acts as generalized truth values object with universal mono $\text{true}: 1 \rightarrow \Omega$ (classifying map)

Role of subobject classifier

- Subobject classifier Ω generalizes set of truth values allowing for internal logic within topos (intuitionistic logic)
- Characteristic functions uniquely map subobjects to Ω representing membership (indicator functions)
- Classification establishes correspondence between subobjects of X and morphisms $X \rightarrow \Omega$ (power set)
- Power objects represent all subobjects of given object defined using subobject classifier ($\wp(X) \cong \Omega^X$)
- Elements of Ω correspond to truth values in internal logic (generalized Boolean algebra)

Structure and Logic of Elementary Topoi

Limits and colimits in topos

- Finite limits include terminal object 1 binary products $A \times B$ equalizers and constructed pullbacks
- Finite colimits include initial object 0 binary coproducts $A + B$ coequalizers and constructed pushouts
- Universality ensures limits and colimits satisfy universal properties (uniqueness up to isomorphism)
- Preservation dictates limits preserved by right adjoints colimits by left adjoints (adjoint functor theorem)

- Interaction with exponentials shows products distribute over colimits exponentials preserve limits in second argument $(A \times (B + C) \cong (A \times B) + (A \times C))$

Internal logic vs categorical structure

- Internal logic of topos corresponds to higher-order intuitionistic type theory (Curry-Howard correspondence)
- Objects represent types morphisms represent functions between types in internal language (lambda calculus)
- Subobject classifier Ω represents proposition type power objects Ω^X represent predicate types on X
- Exponentials Y^X represent function types from X to Y (currying)
- Natural number object provides internal arithmetic (Peano axioms)
- Each topos models its internal logic (categorical semantics)
- Mitchell-Bénabou language formally captures internal logic (typed lambda calculus)
- Kripke-Joyal semantics interprets internal language in terms of topos structure (forcing)

7.2 Set-based topoi and finite topoi

Set-based topoi form the foundation of topos theory. They include familiar categories like sets, functor categories, and presheaf categories. These examples showcase how topoi generalize set-theoretic concepts to broader mathematical structures.

Finite topoi, with limited objects and morphisms, offer simplified versions of topos structures. They maintain key properties like completeness and cocompleteness, while their subobject classifiers constrain internal logic. This connects topos theory to group representations and finite structures.

Set-Based Topoi

Examples of set-based topoi

- Category of sets ($\mathbf{Set}$) forms fundamental example objects are sets morphisms represent functions between sets
- Functor categories $\mathbf{Set}^C$ where C is a small category objects comprise functors from C to $\mathbf{Set}$ morphisms consist of natural transformations (ex: category of graphs)
- Presheaf categories $\mathbf{Set}^{C^{op}}$ where C is a small category objects encompass contravariant functors from C to $\mathbf{Set}$ morphisms involve natural transformations (ex: simplicial sets)
- Slice categories $\mathbf{Set}/X$ where X is a fixed set objects include functions with codomain X morphisms form commutative triangles (ex: pointed sets)

Properties of finite topoi

- Definition topos with finitely many objects and morphisms limited structure
- Examples include finite sets category of sets with n elements or fewer finite groups category of G -sets for finite group G
- Finitely complete and cocomplete all objects possess finite limits and colimits

- Subobject classifier manifests as finite object constrains internal logic
- Representation theory connection finite topoi relate to representations of finite groups illuminate group actions

Sets as elementary topoi

- Finite limits terminal object singleton set binary products cartesian product of sets equalizers subset of elements where functions agree
- Exponentials B^A set of functions from A to B enables function spaces
- Subobject classifier $\Omega = \{0, 1\}$ (false, true) characteristic functions classify subsets
- Cartesian closure isomorphism $Hom(A \times B, C) \cong Hom(A, C^B)$ facilitates currying
- Power objects $P(A)$ power set of A enables set-theoretic operations

Subobject classifier in topoi

- Set-based topoi $\Omega = \{0, 1\}$ true morphism $1 \rightarrow \Omega$ maps to 1 characteristic functions classify subobjects
- Finite topoi subobject classifier finite object structure depends on specific topoi
- Properties classifies subobjects uniquely allows definition of internal logic
- Examples in finite topoi category of G-sets for finite group G Ω is set of G-invariant subsets of G category of n-element sets Ω has n + 1 elements

7.3 Presheaf topoi and functor categories

Presheaf topoi and functor categories are powerful tools in category theory. They generalize the concept of sets and functions, allowing us to work with more complex mathematical structures. These constructions provide a rich framework for studying relationships between objects and morphisms.

Presheaf topoi offer a way to model dynamic information, while functor categories extend this idea to arbitrary categories. Both possess key properties of elementary topoi, including subobject classifiers and internal logic, making them versatile for various mathematical applications.

Presheaf Topoi and Functor Categories

Definition of presheaf topoi

- Presheaf maps objects and morphisms in small category C to sets and functions in Set preserves composition and identity morphisms in opposite direction (functors $F : C^{op} \rightarrow Set$)
- Presheaf topos encompasses presheaves on C as objects and natural transformations between presheaves as morphisms (category $Set^{C^{op}}$)
- Properties of presheaf topos include cartesian closed structure allows internal function objects supports all limits and colimits enables universal constructions possesses subobject classifier provides rich logical framework (Heyting algebra)

Functor categories as elementary topoi

- Functor category comprises functors as objects and natural transformations as morphisms generalizes presheaf topoi to arbitrary source and target categories

- Elementary topos properties demonstrated in functor categories: 1. Construct finite products componentwise 2. Form exponential objects using natural transformations 3. Identify terminal object as constant functor 4. Build equalizers pointwise 5. Define subobject classifier using sieves
- Proof establishes functor categories as models of intuitionistic type theory with dependent types

Subobject classifier in presheaf topoi

- Subobject classifier Ω in $\mathbf{Set}^{C^{op}}$ assigns set of sieves on c to each object c in C (sieves: collections of morphisms with codomain c closed under precomposition)
- True morphism $t: 1 \rightarrow \Omega$ maps terminal object to maximal sieve represents "truth" in the topos
- Characteristic morphism $\chi_m: X \rightarrow \Omega$ for subobject $m: Y \rightarrow X$ encodes membership information for each element x in $X(c)$ $\chi_m(c)(x)$ yields sieve of all $f: d \rightarrow c$ such that $X(f)(x)$ lies in image of $m(d)$
- Construction generalizes power set operation in $\mathbf{Set}$ captures notion of "subset" in presheaf context

Presheaf vs set-based topoi

- Similarities include elementary topos structure subobject classifiers cartesian closed property support for internal logic
- Differences:
 - Presheaf objects: functors encoding both static and dynamic information (sheaves)
 - Set-based objects: sets with additional structure (topological spaces)
 - Subobject classifier: complex in presheaves (sieves) simpler in sets (two-element set)
 - Internal logic: intuitionistic in presheaves classical in sets
 - Presheaf topoi offer nuanced representation of mathematical structures (schemes in algebraic geometry) while set-based topoi provide concrete models (smooth manifolds in differential geometry)
 - Applications: presheaf topoi in sheaf theory and categorical logic set-based topoi in axiomatic set theory and topos-theoretic approaches to quantum mechanics

Unit 8 – Sheaves and Sheafification

8.1 Presheaves and their properties

Presheaves on topological spaces bridge local and global properties, assigning sets to open sets and defining restriction maps. They're key to understanding how information flows between different scales in a space.

Presheaves form a rich category with morphisms, properties like gluing and locality, and diverse examples. This structure lays the groundwork for sheaves and their role in geometry and topology.

Presheaves on Topological Spaces

Definition of presheaves and morphisms

-

Presheaf definition assigns contravariant functor $F : \mathcal{O}(X)^{op} \rightarrow \mathbf{Set}$ maps open sets of topological space X to category of sets

-

Components of a presheaf map open sets $U \subseteq X$ to sets $F(U)$ and inclusions $V \subseteq U$ to restriction maps $\rho_{U,V} : F(U) \rightarrow F(V)$

-

Morphisms between presheaves use natural transformations consisting of maps $\eta_U : F(U) \rightarrow G(U)$ compatible with restriction maps $\eta_V \circ \rho_{U,V}^F = \rho_{U,V}^G \circ \eta_U$

Properties of presheaves

-

Restriction maps encode local-to-global relationship with functoriality $\rho_{U,U} = id_{F(U)}$ and $\rho_{V,W} \circ \rho_{U,V} = \rho_{U,W}$ for $W \subseteq V \subseteq U$

-

Gluing axiom allows unique section $s \in F(U)$ from agreeing sections $s_i \in F(U_i)$ on open cover $\{U_i\}$ of U

-

Locality property determines sections by restrictions to smaller open sets

Examples of presheaves

-

Presheaf of continuous functions maps $F(U) = C(U, \mathbb{R})$ with function restriction maps (real-valued functions)

-

Constant presheaf assigns fixed set A to all open sets U with identity restriction maps

-

Presheaf of bounded functions uses set of bounded real-valued functions on U with function restrictions

-

Presheaf of local constants employs set of locally constant functions on U with function restrictions

Category of presheaves

-

Category of presheaves uses presheaves as objects and natural transformations as morphisms

-

Properties include completeness, cocompleteness, abelian category structure, and topos structure

-

Yoneda embedding functor $\gamma : \mathcal{O}(X) \rightarrow \mathbf{PSh}(X)$ maps open sets to representable presheaves

-

Subobject classifier Ω presheaf of local truth values maps $\Omega(U)$ to set of open subsets of U

-

Exponential objects G^F exist for presheaves F and G with $(G^F)(U)$ as set of natural transformations $F|_U \rightarrow G|_U$

8.2 Definition and examples of sheaves

Sheaves are mathematical structures that connect local and global information on topological spaces. They extend presheaves by adding axioms for local identity and global consistency, allowing us to model complex relationships between data on different scales.

Sheaf theory finds applications across mathematics, from algebraic geometry to topology. By providing a framework for gluing local data into global structures, sheaves enable powerful tools like sheaf cohomology and play a crucial role in modern mathematical research.

Sheaf Theory Fundamentals

Definition of sheaves

- Presheaf maps open sets of topological space to sets, reverses inclusion relations
- Sheaf axioms ensure local identity and global consistency of sections
- Sheaves satisfy both locality and gluing axioms, extending presheaf definition
- Category theory defines sheaves using limits in presheaf category (equalizer diagrams)

Local nature of sheaves

- Sheaves model local-to-global properties, define information on open sets
- Gluing axiom combines compatible local sections into global sections

- Local data consistency crucial for mathematical modeling (differential equations)

Examples of sheaves

- Continuous functions on topological space X map open sets to continuous functions
- Smooth functions on manifolds associate smooth functions to open subsets
- Vector bundle sections provide local trivializations over base space
- Constant sheaf assigns fixed set to all open subsets (real numbers)
- Étale space geometrically represents sheaf (stalks as fibers)

Sheaves vs local homeomorphisms

- Étale space forms total space of sheaf, locally homeomorphic to base
- Sheaves reconstructed from étale spaces, sections correspond to sheaf sections
- Sheafification transforms presheaves to sheaves using étale space construction

Importance of sheaves in mathematics

- Algebraic geometry uses structure sheaves for varieties, coherent sheaves on schemes
- Differential geometry employs differential form sheaves, tangent/cotangent sheaves
- Complex analysis utilizes holomorphic function sheaves (Riemann surfaces)
- Topology develops sheaf cohomology, derived functors (homological algebra)
- Mathematical physics applies sheaves in quantum field theory, D-modules
- Representation theory uses sheaves of group representations (Lie groups)
- Categorical perspective generalizes to Grothendieck topologies, topos theory

8.3 Sheafification and associated sheaf functor

Sheafification is a process that transforms presheaves into sheaves, bridging local and global information. It's crucial for studying geometric structures like manifolds and fiber bundles, ensuring coherent global structure while preserving local properties.

The sheafification functor uses a two-step process: plus construction and iteration. This method creates a sheaf from a presheaf, maintaining important properties and establishing a universal property. It's widely used in various mathematical applications.

Sheafification Process and Functor

Concept of sheafification

- Presheaf defined as contravariant functor from topological space to category of sets fails to satisfy sheaf axioms
- Sheaf defined as presheaf satisfying locality and gluing conditions ensures coherent global structure

- Sheafification process minimally modifies presheaf to create sheaf preserves local properties while adding global coherence
- Motivation bridges local and global information essential for studying geometric and topological structures (manifolds, fiber bundles)

Construction of sheafification functor

- Two-step process involves plus construction and iteration
- Plus construction defines sections over open sets as compatible families

$$F^+(U) = \{(s_i)_{i \in I} \mid s_i \in F(U_i), s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}\}$$
- Iteration applies plus construction twice F^{++} results in a sheaf
- Universal property establishes natural transformation from presheaf to its sheafification unique up to isomorphism
- Functorial nature preserves morphisms between presheaves commutes with restrictions to open subsets

Applications of sheafification process

- Constant presheaf sheafification yields locally constant sheaf (étalé spaces)
- Presheaf of continuous functions already a sheaf sheafification is identity
- Presheaf of bounded functions sheafification gives sheaf of locally bounded functions
- Presheaf of rational functions sheafification results in sheaf of meromorphic functions (complex analysis)
- Stalk-wise sheafification constructs sheaf from presheaf by considering stalks at each point

Presheaves vs sheaves categories

- Adjunction between categories establishes sheafification functor as left adjoint to inclusion functor
- Reflection of sheaves in presheaves positions sheafification as reflector
- Sheafification functor preserves colimits maintains structural relationships
- Effect on morphisms induces bijection between morphism sets in certain cases
- Sheafification as idempotent operation applying sheafification to a sheaf yields isomorphic sheaf
- Connection to site theory generalizes to Grothendieck topologies expands applicability

Unit 9 – Grothendieck Topoi

9.1 Sites and Grothendieck topologies

Sites and Grothendieck topologies expand on the idea of open covers in topology. They provide a framework for studying geometric and algebraic objects in a unified way, allowing us to define "local" properties in diverse mathematical contexts.

The key components of a site include an underlying category and a collection of covering families. These define local structure and enable the study of sheaves, which encapsulate local-to-global data. This approach is crucial for understanding modern algebraic geometry.

Sites and Grothendieck Topologies

Concept and examples of sites

- Site definition combines category $\mathcal{C}$ with Grothendieck topology generalizing topological spaces
- Examples of sites encompass small category of open sets in topological space, Zariski site in algebraic geometry, Étale site for schemes, Crystalline site in p-adic cohomology
- Key components of a site include underlying category and collection of covering families defining local structure
- Sites provide framework for studying geometric and algebraic objects in unified manner
- Applications extend to sheaf theory, cohomology theories, and algebraic geometry

Axioms of Grothendieck topology

- Grothendieck topology assigns "covering families" to objects in category formalizing local-to-global principle
- Axioms ensure consistency and compatibility of covering families: 1. Identity axiom requires singleton family containing identity morphism to be a covering 2. Stability under base change allows pullback of coverings along morphisms 3. Transitivity (local character) enables composition of coverings
- Properties include refinement of coverings allowing finer local descriptions and intersection of coverings capturing common local information
- Generalizes open cover concept from classical topology to abstract categorical setting
- Enables study of "local" properties in diverse mathematical contexts (algebraic varieties, schemes)

Construction of sheaf categories

- Presheaf on site defined as contravariant functor from site to Set category encapsulating local-to-global data
- Sheaf condition requires local data to glue consistently with respect to covering families
- Construction process involves: 1. Starting with category of presheaves 2. Identifying presheaves satisfying sheaf condition 3. Forming full subcategory of sheaves
- Yoneda embedding for sites represents objects as presheaves preserving site structure

- Sheafification functor transforms presheaves into sheaves acting as left adjoint to inclusion
- Allows transition from local to global perspective in geometric and algebraic contexts

Sheaves as topos formation

- Topos defined as category equivalent to category of sheaves on a site providing rich structural framework
- Key properties to verify for topos structure:
 - Existence of finite limits and colimits (pullbacks, pushouts)
 - Subobject classifier (truth object) generalizing characteristic functions
 - Power objects representing "set" of subobjects
- Proof outline demonstrates category of sheaves is complete, cocomplete, and possesses necessary categorical structures
- Site-dependent constructions highlight flexibility of topos theory in capturing diverse mathematical contexts
- Consequences of topos structure include internal logic for reasoning and geometric morphisms for relating toposes
- Provides unified approach to geometry, algebra, and logic in abstract categorical setting

9.2 Definition and properties of Grothendieck topoi

Grothendieck topoi generalize spaces using categories of sheaves. They're defined by Giraud axioms, which ensure properties like having all small colimits and effective equivalence relations. These axioms allow for complex object construction and quotient operations.

Grothendieck topoi have key properties like local cartesian closure and a subobject classifier. These enable internal logic, type theory, and representation of truth values. Topoi also provide a framework for descent theory, generalizing the sheaf condition for gluing local data.

Definition and Properties of Grothendieck Topoi

Definition of Grothendieck topos

- Category equivalent to category of sheaves on a site allows for generalized notion of space
- Characterized by Giraud axioms provide necessary and sufficient conditions:
 - Has all small colimits enables construction of complex objects
 - Filtered colimits commute with finite limits ensures compatibility of operations
 - Equivalence relations are effective allows for quotient constructions
 - Has a small set of generators provides a way to build all objects

Properties of Grothendieck topos

- Local cartesian closure enables internal logic and type theory
- Slice category $\mathcal{E}/A$ cartesian closed for any object A

- Exponentials exist in slice categories allows for function spaces
- Pullback functors have right adjoints ensures good behavior of base change
- Subobject classifier Ω represents truth values and subobjects
- Characteristic functions of subobjects factor through Ω
- Defined by universal property with terminal object 1 and monic arrow $true : 1 \rightarrow \Omega$
- For any monic $m : S \rightarrow X$, unique $\chi_m : X \rightarrow \Omega$ makes pullback square

Grothendieck topoi and descent

- Descent theory studies gluing local data to obtain global objects (vector bundles, quasi-coherent sheaves)
- Topoi provide framework for descent generalizing sheaf condition
- Sheaves in topos satisfy descent conditions ensuring global consistency
- Descent data corresponds to compatible local sections
- Categorical formulation uses slice categories and pullback functors

Sheaves on Heyting algebra

- Complete Heyting algebra has all meets and joins, satisfies infinite distributive law
- Category of sheaves consists of order-preserving functions satisfying sheaf condition
- Proof of Grothendieck topos structure: 1. Verify Giraud axioms 2. Construct subobject classifier 3. Show cartesian closure
- Key steps involve using Heyting algebra properties to define sheaf operations and exploiting relationship between opens in topology and algebra elements

9.3 Comparison with elementary topoi

Grothendieck and elementary topoi are powerful mathematical structures that generalize the notion of space and set theory. They share key properties like having a subobject classifier and being cartesian closed, enabling internal logic and higher-order functions.

However, they differ in size, completeness, and existence of generators. Grothendieck topoi are typically larger and more complete, while elementary topoi are more flexible in size. These differences impact their applications in mathematics and logic.

Grothendieck Topoi vs Elementary Topoi

Properties of Grothendieck vs elementary topoi

- Grothendieck topoi
- Categories equivalent to categories of sheaves on a site defined as mathematical structures generalizing notion of topological space (étale topology)
- Properties:
- Possess all small limits and colimits enabling wide range of categorical constructions

- Have a subobject classifier allowing for internal logic and set-theoretic operations
- Are cartesian closed supporting function spaces and higher-order functions
- Elementary topoi
 - Categories satisfying axioms resembling properties of category of sets formulated to capture essential features of set theory categorically
- Properties:
 - Have finite limits and colimits supporting basic categorical operations (products, equalizers)
 - Possess a subobject classifier enabling internal logic similar to Grothendieck topoi
 - Are cartesian closed allowing for function spaces and higher-order functions
- Key differences
- Size considerations
 - Grothendieck topoi typically large categories containing infinite objects (sheaves on infinite spaces)
 - Elementary topoi can be small or large encompassing finite and infinite structures
- Completeness
 - Grothendieck topoi complete and cocomplete allowing for arbitrary limits and colimits
 - Elementary topoi only require finite limits and colimits restricting some constructions
- Existence of generators
 - Grothendieck topoi have a set of generators enabling representation of objects in terms of simpler ones
 - Elementary topoi do not necessarily have generators lacking this structural property

Examples of non-overlapping topoi

- Grothendieck topoi not elementary topoi
 - Category of sheaves on non-trivial topological space (manifolds, algebraic varieties)
 - Category of G -sets for infinite group G (representations of infinite symmetry groups)
 - Category of sheaves on non-trivial Grothendieck site (étale site in algebraic geometry)
- Elementary topoi not Grothendieck topoi
 - Effective topos modeling realizability in computer science
 - Category of finite sets lacking infinite objects required for Grothendieck topoi
 - Category of nominal sets used in theoretical computer science for name-binding

Axiom of choice in topoi distinction

- Grothendieck topoi
 - Always holds due to existence of enough projectives ensuring choice functions

- Enables powerful results in algebraic geometry and topology
- Elementary topoi
- May or may not hold depending on specific topos
- Allows construction of counterexamples to properties valid in Grothendieck topoi (non-well-orderable objects)
- Implications for model theory
- Grothendieck topoi model intuitionistic higher-order logic with choice supporting classical mathematics
- Elementary topoi model intuitionistic higher-order logic without choice enabling constructive approaches

Foundations and Logic

Implications for mathematics and logic

- Foundations of mathematics
- Generalized notion of space extending beyond traditional topological spaces
- Framework for synthetic differential geometry allowing infinitesimals
- Constructive mathematics development without law of excluded middle
- Higher-order logic
- Models for intuitionistic type theory supporting dependent types
- Internal languages of topoi enabling categorical logic
- Categorical logic study formalizing mathematical reasoning
- Applications in other areas
- Algebraic geometry: étale cohomology revolutionizing study of schemes
- Sheaf theory: generalized sheaves on sites beyond topological spaces
- Category theory: generalizing set-theoretic foundations to categorical setting
- Philosophical implications
- Questions set theory primacy as foundation suggesting alternatives
- Structural approach to mathematics emphasizing relationships over objects
- Highlights category theory importance in foundations unifying diverse areas

Unit 10 – Geometric Morphisms and Classifying Topoi

10.1 Definition and properties of geometric morphisms

Geometric morphisms are special functors between topoi that preserve key structures. They consist of an inverse image functor and a direct image functor, forming an adjoint pair with specific preservation properties.

These morphisms play a crucial role in connecting different topoi, allowing us to transfer information and structures between them. Examples include relationships between sets, presheaves, and sheaves, highlighting their importance in various mathematical contexts.

Geometric Morphisms in Topos Theory

Definition of geometric morphisms

- Geometric morphism comprises functor $f^* : \mathcal{F} \rightarrow \mathcal{E}$ between topoi with right adjoint $f_* : \mathcal{E} \rightarrow \mathcal{F}$, where f^* preserves finite limits (pullbacks, terminal objects)
- Composition of geometric morphisms yields another geometric morphism, allowing chaining of morphisms between multiple topoi
- Identity functor serves as geometric morphism, providing reflexivity in topos relationships
- Inverse image functor f^* and direct image functor f_* form core components, each playing distinct roles in structure preservation
- Adjunction relationship $f^* \dashv f_*$ established through natural isomorphism $\text{Hom}_{\mathcal{E}}(f^*(A), B) \cong \text{Hom}_{\mathcal{F}}(A, f_*(B))$, connecting morphisms in source and target topoi

Preservation of categorical structures

- f^* preserves finite limits by definition, crucial for maintaining topos structure (terminal objects, pullbacks)
- f^* preserves finite colimits due to left adjoint property, extends to preservation of initial objects and pushouts
- Exponential preservation by f^* shown through isomorphism $f^*(B^A) \cong (f^*B)^{f^*A}$, utilizing adjunction and limit preservation
- Proofs involve demonstrating preservation of specific constructions (limits, colimits, exponentials) using functor properties and adjunction relationships

Examples in common topoi

- Set and topos $\mathcal{E}$ connected by global sections functor $\Gamma : \mathcal{E} \rightarrow \text{Set}$ and constant objects functor $\Delta : \text{Set} \rightarrow \mathcal{E}$
- Presheaf topoi linked through functors between base categories, employing left and right Kan extensions

- Subtopos inclusion uses inclusion functor as direct image and sheafification functor as inverse image
- Sheaf topoi connected via continuous functions between topological spaces, utilizing direct and inverse image sheaf functors (restriction, extension)

Relation to adjoint functors

- Geometric morphisms form special adjoint pair $f^* \dashv f_*$ with additional finite limit preservation condition
- Left adjoint f^* preserves colimits while right adjoint f_* preserves limits, fundamental to structure preservation
- Not all topos adjunctions qualify as geometric morphisms, highlighting unique nature of these morphisms
- Frobenius reciprocity $f^*(A) \times_{\mathcal{E}} B \cong f^*(A \times_{f_*} B)$ characterizes special interaction between inverse image and fiber products
- Topos equivalence occurs when both f^* and f_* are equivalences, preserving all topos-theoretic properties (subobject classifier, power objects)

10.2 Classifying topoi and universal properties

Classifying topoi are powerful tools in category theory, representing models of geometric theories as objects with universal properties. They provide a way to understand and classify mathematical structures through the lens of topos theory.

The existence and uniqueness of classifying topoi are proven through syntactic categories and sheafification. Examples include Sets, Groups, and Rings theories. Applications involve using universal properties to solve classification problems in various mathematical domains.

Classifying Topoi

Concept of classifying topos

- Classifying topos represents models of a geometric theory as category-theoretic object exhibiting universal property for geometric morphisms
- Geometric theory consists of first-order axioms with syntactic restrictions allowing geometric constructions (intersection, union, existential quantification)
- Models of geometric theory comprise set-theoretic structures satisfying axioms with morphisms preserving structure (homomorphisms)

Existence of classifying topoi

- Existence proven by constructing syntactic categories and applying sheafification process on syntactic site
- Uniqueness demonstrated using universal property and category equivalence argument
- Proof involves: 1. Define appropriate Grothendieck topology 2. Show resulting topos satisfies universal property 3. Establish equivalence between constructed topos and any other satisfying the property

Examples of classifying topoi

- Sets theory classifying topos category of sets (Set)
- Groups theory classifying topos category of group actions (BG)
- Rings theory classifying topos category of ring actions
- Local rings theory classifying topos Zariski topos
- Construction process: 1. Identify geometric theory 2. Determine appropriate site 3. Apply sheafification process

Applications of classifying topoi

- Universal property establishes natural bijection between geometric morphisms and models
 $Hom(E, Set[T]) \cong Mod_T(E)$
- Geometric morphisms classification corresponds to models of theory in codomain topos
- Classification problems solved by: 1. Identify relevant geometric theory 2. Construct or recognize classifying topos 3. Use universal property to establish bijections
- Applications include classifying topological spaces, algebraic structures (groups, rings), studying model-theoretic properties

10.3 Applications to algebraic geometry and model theory

Classifying topoi bridge algebraic geometry and model theory, unifying diverse mathematical structures. They encode geometric objects like schemes and algebraic spaces as models of theories, revealing deep connections between algebra, geometry, and logic.

These topoi provide a powerful framework for studying mathematical theories and their models. By representing theories as categories of sheaves, they offer new insights into the nature of mathematical structures and their relationships.

Classifying Topoi in Algebraic Geometry and Model Theory

Schemes as classifying topoi

- Schemes generalize algebraic varieties unify affine and projective varieties
- Locally ringed spaces glue together affine schemes ($\text{Spec} A$, $\text{OSpec} A$) where A is a commutative ring
- Geometric theory of schemes involves locally ringed spaces satisfying specific axioms (gluing, local nature)
- Zariski topology on schemes arises from prime ideals of rings reflects algebraic properties geometrically
- Classifying topos of a scheme encodes points as models of its geometric theory (Spec functor)

Algebraic spaces as classifying topoi

- Algebraic spaces generalize schemes allow for quotients by group actions not representable by schemes

- Étale topology refines Zariski topology captures local properties more effectively (formal étaleness)
- Sheaves on the étale site of a scheme define algebraic spaces (étale equivalence relations)
- Geometric theory of algebraic spaces involves étale descent conditions and local properties
- Classifying topos of an algebraic space represents models of its associated geometric theory

Existence of classifying topos

- Coherent theories use finitary logic allow finite conjunctions, disjunctions, and existential quantification
- Geometric theories extend coherent theories include infinitary disjunctions capture more general properties
- Syntactic site construction: 1. Form category of formulas and provable implications 2. Define Grothendieck topology using covering families 3. Take sheaves on this site as the classifying topos
- Universal model in classifying topos represents generic model of the theory
- Completeness theorem: every model of the theory corresponds to a point of the classifying topos
- Soundness theorem: logical consequences in the theory reflected by geometric morphisms between topos

Classifying topos in geometry vs model theory

- Geometric morphisms between classifying topos correspond to interpretations between theories
- Varieties as models of geometric theories (equations, inequations in polynomial rings)
- Zariski spectra of rings form classifying topos for theories of local rings with given global sections
- Categoricity in model theory relates to properties of classifying topos (e.g., two-valued)
- Stability theory connects to topos-theoretic notions (e.g., coherent objects in the classifying topos)
- Stone duality generalizes to classify Boolean algebras, Heyting algebras, and certain topos

Unifying role of classifying topos

- Topos theory provides common language for algebraic geometry and model theory (sites, sheaves, geometric morphisms)
- Grothendieck topologies generalize classical topological spaces unify various notions of "space" (étale, fppf, crystalline)
- Categorical logic interprets mathematical theories within topos allows for alternative foundations
- Synthetic differential geometry develops smooth infinitesimal analysis in certain topos (well-adapted models)
- Applications extend to algebraic topology (homotopy types as ∞ -topos), functional analysis (C^* -algebras as ringed topos)

Unit 11 – Internal Logic and Mitchell–Bénabou Language

11.1 Internal language of a topos

The internal language of a topos is a powerful tool that bridges category theory and logic. It uses higher-order intuitionistic logic to express properties within a topos, allowing for intuitive reasoning about complex structures.

This language consists of types, terms, logical connectives, and quantifiers. It's interpreted through Kripke-Joyal semantics, linking logical formulas to subobjects in the topos. This framework enables proofs and deeper understanding of topos theory.

Foundations of Internal Language in Topos Theory

Internal language of topos

- Formal language expresses properties and relationships within topos using higher-order intuitionistic logic
- Components include types corresponding to topos objects, terms representing morphisms or object elements, logical connectives (and, or, implies, not), and quantifiers (for all, there exists)
- Allows intuitive reasoning about topos structures bridging gap between category theory and logic facilitating proofs of theorems within topos framework
- Kripke-Joyal semantics interprets internal language statements in topos linking logical formulas to subobjects

Construction of well-formed formulae

- Atomic formulas establish equality between terms while compound formulas use logical connectives and quantifiers
- Logical connectives: conjunction ($\phi \wedge \psi$), disjunction ($\phi \vee \psi$), implication ($\phi \Rightarrow \psi$), negation ($\neg \phi$)
- Quantifiers: universal ($\forall x : A. \phi(x)$), existential ($\exists x : A. \phi(x)$)
- Type theory notation ($x : A$) denotes x as term of type A
- Formation rules ensure syntactic correctness of formulas specifying how to combine terms and formulas

Interpretation in specific topos

- Maps internal language constructs to topos structures evaluating truth values of formulas
- Context-dependent interpretation means same formula may have different meanings in different toposes
- Subobject classifier plays crucial role in interpreting logical operations representing truth values in topos

- Generalized elements used to interpret quantified statements correspond to morphisms in topos
- Sheaf semantics interprets formulas in terms of local sections crucial for Grothendieck toposes

Proofs using internal language

- Strategies involve translating categorical statements into internal language applying logical inference rules and interpreting results back in categorical terms
- Key techniques include natural deduction in intuitionistic logic and use of characteristic morphisms for subobject relations
- Mitchell-Bénabou language extends internal language for more expressive proofs including lambda abstraction and application
- Soundness and completeness ensure validity of proofs in internal language relating provability to truth in topos
- Applications prove properties of specific toposes (Sets, Sheaves) and establish general results about classes of toposes

11.2 Mitchell-Bénabou language and its semantics

The Mitchell-Bénabou language provides a formal system for expressing concepts in topos theory. It uses types, terms, and formulas to represent objects, morphisms, and relationships within a topos, allowing for precise mathematical reasoning.

The language's syntax and semantics bridge the gap between abstract category theory and concrete logical statements. This enables mathematicians to work with topos-theoretic ideas using familiar logical constructs while maintaining rigorous foundations.

Syntax and Semantics of the Mitchell-Bénabou Language

Syntax of Mitchell-Bénabou language

- Basic components of the language form foundation for expressing topos-theoretic concepts
- Types correspond to objects in topos (natural numbers, real numbers)
- Terms represent morphisms or elements of objects
- Formulas express properties and relationships between objects
- Type construction builds complex types from simpler ones
- Base types directly correspond to objects in topos
- Product types combine multiple types ($A \times B$)
- Power types represent subobject classifiers (Ω^A)
- Term formation rules define valid terms in the language
- Variables stand for arbitrary elements of a type
- Constants represent specific elements or morphisms
- Function application applies a function to arguments ($f(x)$)

- Lambda abstraction defines functions ($\lambda x. t$)
- Formula construction creates logical statements
- Atomic formulas express basic predicates ($x = y, P(x)$)
- Logical connectives combine formulas ($\wedge, \vee, \Rightarrow$)
- Quantifiers express universal and existential statements ($\forall x, \exists x$)
- Typing rules ensure well-formedness of expressions
- Well-formed terms have consistent types
- Type inference determines types of complex expressions

Semantics in topos logic

- Interpretation of types maps language constructs to topos objects
- Objects in topos provide semantic domain for types
- Subobject classifier Ω interprets truth values
- Semantics of terms define meaning of expressions
- Global sections represent constant terms
- Morphisms in topos interpret function terms
- Truth values in topos logic generalize classical Boolean logic
- Subobject classifier Ω replaces two-element Boolean algebra
- Characteristic functions map subobjects to truth values
- Interpretation of logical connectives uses topos structure
- Meet and join in subobject lattice interpret $\wedge$ and $\vee$
- Heyting algebra structure provides intuitionistic logic semantics
- Quantifier semantics leverage adjoint functors
- Universal quantification uses right adjoint to pullback
- Existential quantification uses left adjoint to pullback
- Soundness and completeness relate syntax and semantics
- Provable statements are valid in all models
- Valid statements are provable in the formal system

Translation between internal languages

- Identifying corresponding constructs bridges different representations
- Objects and types have direct correspondence
- Morphisms translate to terms in Mitchell-Bénabou language

- Subobjects become predicates in logical formulas
- Translation techniques preserve meaning across languages
- Internal language to Mitchell-Bénabou involves formalizing intuitive concepts
- Mitchell-Bénabou to internal language requires interpreting formal syntax
- Handling quantifiers requires careful treatment
- Bounded quantification restricts quantifiers to specific domains
- Dependent types express relationships between types
- Preserving logical structure maintains equivalence of statements
- Conjunctions and disjunctions translate directly
- Implications and negations may require adjustment for intuitionistic logic

Reasoning with Mitchell-Bénabou language

- Expressing categorical concepts formalizes topos theory
- Monomorphisms and epimorphisms defined via injective and surjective properties
- Pullbacks and pushouts described using universal properties
- Limits and colimits expressed as terminal and initial objects
- Defining and working with natural number objects captures arithmetic
- Peano axioms formalized in Mitchell-Bénabou language
- Recursion and induction principles enable proofs about natural numbers
- Reasoning about subobjects leverages topos structure
- Membership and inclusion expressed using characteristic functions
- Operations on subobjects (intersection, union) defined via logical connectives
- Expressing universal properties captures essence of categorical constructions
- Products and coproducts defined using projections and injections
- Exponential objects formalized with evaluation and currying
- Working with subobject classifier enables powerful reasoning
- Characteristic functions map subobjects to truth values
- Power objects represent collections of subobjects
- Formalizing topos-specific concepts captures unique features
- Cartesian closed structure expressed using exponential objects
- Local cartesian closure formalized using dependent product types

11.3 Kripke-Joyal semantics and forcing

Kripke-Joyal semantics in topos theory interprets statements in the topos internal language. It uses forcing conditions to determine the validity of propositions, reflecting the topos structure and properties when evaluating truth values.

This approach generalizes Kripke semantics for intuitionistic logic. Truth values are represented by subobjects of the terminal object in a topos, forming a Heyting algebra structure that extends beyond classical Boolean logic.

Foundations of Kripke-Joyal Semantics

Kripke-Joyal semantics in topos theory

- Formal system interprets statements in topos internal language generalizing Kripke semantics for intuitionistic logic
- Reflects topos structure and properties evaluating truth values of statements
- Utilizes forcing conditions to determine validity of propositions (independence of continuum hypothesis)

Truth values in internal language

- Represented by subobjects of terminal object in topos
- Form Heyting algebra structure generalizing Boolean algebra
- Can be viewed as open sets in topological space (sheaf-theoretic perspective)

Forcing in internal language

- Technique determines truth values of statements adapted from set theory
- Applies forcing conditions to evaluate complex formulas
- Steps: 1. Identify relevant forcing conditions 2. Evaluate truth values under these conditions 3. Combine results to determine overall truth value
- Used for proving independence of axioms and constructing models with specific properties (non-standard models of arithmetic)

Kripke-Joyal vs classical semantics

- Classical semantics uses two-valued logic (true/false) and law of excluded middle
- Kripke-Joyal based on intuitionistic logic with more complex truth values (Heyting algebra)
- Law of excluded middle and double negation elimination not generally valid in Kripke-Joyal
- Both provide formal systems for evaluating truth and share common logical connectives and quantifiers

Applications of Kripke-Joyal semantics

- Problem-solving approach: 1. Formulate problem in topos internal language 2. Identify relevant forcing conditions 3. Apply semantics to evaluate truth values

- Proves properties of objects within topos and establishes relationships between different topoi
- Used in algebraic geometry for sheaf theory and descent theory
- Applied in computer science for domain theory and programming language semantics

Unit 12 – Topos Theory: Algebraic and Geometric Views

12.1 Algebraic theories in topoi

Algebraic theories in topoi provide a powerful framework for studying mathematical structures. They allow us to describe and analyze groups, rings, and other algebraic objects within a generalized universe of sets, offering new insights and perspectives.

Free models play a crucial role in this context, embodying universal properties and facilitating the construction of algebraic structures. The interplay between algebraic and Lawvere theories, along with the unique features of topoi, opens up exciting avenues for mathematical exploration.

Foundations of Algebraic Theories in Topoi

Algebraic theories in topoi

- Algebraic theories serve as formal systems describing algebraic structures encompassing sorts, operations, and equations (groups, rings)
- Models in a topos manifest as objects satisfying theory axioms with morphisms respecting operations
- Operations interpreted as arrows in topos representing functions (addition, multiplication)
- Equations satisfied through commutative diagrams in topos (associativity, distributivity)

Free models in topoi

- Free model concept embodies universal property in category theory acting as initial object in model category
- Construction process utilizes coproducts and quotients in topos, iteratively applying operations
- Adjoint functor pair consists of forgetful functor and free functor
- Free functor as left adjoint preserves limits (products, equalizers)

Connections and Frameworks

Algebraic vs Lawvere theories

- Lawvere theories offer category-theoretic formulation of algebraic theories with objects representing arities and morphisms representing terms
- Equivalence exists between algebraic and Lawvere theories, with models of algebraic theories corresponding to product-preserving functors
- Lawvere theories provide more categorical approach facilitating work in certain contexts (abstract algebra, universal algebra)

Topoi for algebraic structures

- Topos functions as generalized universe of sets with internal logic, subobject classifier, and power objects
- Algebraic structures interpreted as objects in topos (groups, rings, modules)
- Geometric morphisms preserve algebraic structures between topoi
- Sheaf models allow algebraic structures to vary over a space (vector bundles, local rings)
- Synthetic approach enables axiomatization of mathematics within a topos (synthetic differential geometry, smooth infinitesimal analysis)

12.2 Synthetic differential geometry

Synthetic differential geometry revolutionizes calculus by using infinitesimals instead of limits. This approach simplifies complex calculations and provides a more intuitive foundation for advanced mathematical concepts.

The theory's principles, like microlinearity and integration axioms, allow for elegant formulations of geometric ideas. Models in smooth topoi provide consistent interpretations, bridging the gap between intuition and rigorous mathematics.

Foundations of Synthetic Differential Geometry

Principles of synthetic differential geometry

- Infinitesimal elements form basis of calculus without limits
- Nilpotent infinitesimals square to zero ($d^2 = 0$)
- Kock-Lawvere axiom defines unique solutions for certain polynomial equations
- Smooth functions generalize differentiable functions
- Microlinearity ensures local linearity at infinitesimal scales
- Infinitesimal linearity allows Taylor expansions with nilpotent terms
- Axiom of integration enables definition of integral without limits
- Well-adapted models provide consistent interpretations (topos theory)
- Principle of microaffinity guarantees existence of tangent spaces
- Principle of microcancellation allows cancellation of infinitesimals in equations

Models in suitable topoi

- Smooth topoi provide framework for synthetic differential geometry
- Cahiers topos uses sheaves on infinitesimal sites
- Dubuc topos employs C^∞ -rings and ideals
- Germ-determined functions capture local behavior near a point
- Smooth loci generalize notion of smooth manifolds

- Grothendieck topologies define sheaves on categories
- Sheaf models interpret synthetic differential geometry axioms
- Presheaf categories offer flexible settings for constructing models

Applications and Comparisons

Applications to analysis and geometry

- Tangent vectors and bundles defined using infinitesimal paths
- Differential forms represent infinitesimal areas and volumes
- De Rham cohomology computed using nilpotent infinitesimals
- Lie groups and algebras studied with infinitesimal symmetries
- Manifolds in synthetic context generalize smooth spaces
- Calculus of variations solved using infinitesimal perturbations
- Symplectic geometry formulated with infinitesimal phase spaces

Synthetic vs classical differential geometry

- Intuitionistic logic allows infinitesimals, classical logic doesn't
- Smooth infinitesimal analysis provides rigorous foundation
- Treatment of singularities more natural in synthetic approach
- Coordinate-free approach emphasizes intrinsic geometry
- Generalization of manifolds includes singular and infinite-dimensional spaces
- Nilpotent infinitesimals simplify calculations (derivatives, Taylor series)
- Consistency with classical results ensured by transfer principle
- Advantages in certain geometrical constructions (tangent bundles, jet spaces)

12.3 Topological and smooth topoi

Topological and smooth topoi are powerful frameworks in category theory. They provide a way to study spaces and their properties using sheaves, which are collections of local data that fit together consistently.

These topoi have important applications in mathematics and physics. Topological topoi are used in studying continuous spaces, while smooth topoi are crucial for analyzing differentiable structures and infinitesimal methods in geometry.

Topological and Smooth Topoi

Topological Topos

- Definition of a topological topos encapsulates category of sheaves on a topological space, utilizes Grothendieck topology on category of open sets

- Properties of topological topoi include subobject classifier representing truth values, natural number object modeling arithmetic
- Examples of topological topoi encompass sheaves on a topological space representing local data, étale spaces over a topological space capturing local homeomorphisms

Smooth Topos

- Definition of a smooth topos involves category of sheaves on site of smooth manifolds, employs Grothendieck topology on category of smooth manifolds
- Properties of smooth topoi feature subobject classifier in smooth context representing smooth truth values, line object playing crucial role in infinitesimal analysis
- Examples of smooth topoi include sheaves on site of smooth manifolds modeling smooth data, synthetic differential geometry providing foundation for infinitesimal methods

Comparison between Topological and Smooth Topoi

- Similarities highlight both as Grothendieck topoi with well-defined categorical structures, both possess notion of local structure allowing localization of properties
- Differences arise in nature of underlying site (topological spaces vs smooth manifolds), smoothness requirements in smooth topos constraining morphisms
- Functors between topological and smooth topoi comprise forgetful functor from smooth to topological forgetting smooth structure, smooth structure functor adding smoothness when applicable

Applications of Topological and Smooth Topoi

- Topological applications involve cohomology theories studying topological invariants, homotopy theory analyzing continuous deformations
- Smooth applications encompass differential geometry studying smooth manifolds, synthetic differential geometry providing rigorous infinitesimal analysis
- Interdisciplinary applications include algebraic geometry combining algebra and geometry, mathematical physics modeling physical phenomena

Unit 13 – Topos Theory and Foundations of Mathematics

13.1 Intuitionistic logic and constructive mathematics

Intuitionistic logic challenges traditional mathematical thinking by rejecting the law of excluded middle. It demands constructive proofs, providing explicit methods or algorithms, and avoids proof by contradiction. This approach aligns math with computational realizability and addresses foundational issues.

Intuitionistic logic differs from classical logic in its treatment of negation and proof techniques. It forms a subsystem of classical logic, with all intuitionistically valid formulas being classically valid. This shift impacts mathematical practice, set theory, and computational mathematics.

Foundations of Intuitionistic Logic and Constructive Mathematics

Principles of intuitionistic logic

- Law of excluded middle rejected challenges traditional true-false dichotomy
- Truth established through constructive proof provides explicit methods or algorithms
- Proof by contradiction generally avoided limits inference techniques
- Addresses foundational issues in mathematics mitigates set theory paradoxes (Russell's paradox)
- Aligns mathematical practice with computational realizability enhances practical applicability
- Brouwer's intuitionism views mathematics as human mental construct rejects platonic mathematical realism

Intuitionistic vs classical logic

- Negation handled differently double negation doesn't imply affirmation ($\neg\neg A$ weaker than A)
- Proof by contradiction limited not universally applicable as in classical logic
- Law of excluded middle ($A \vee \neg A$) rejected in intuitionistic logic
- Intuitionistic logic forms subsystem of classical logic all intuitionistically valid formulas classically valid
- De Morgan's laws partially valid in intuitionistic logic fully valid in classical logic

Constructive vs non-constructive proofs

- Constructive: Euclidean algorithm for GCD provides step-by-step computation method
- Non-constructive: Existence of irrational numbers proven by contradiction doesn't specify number
- Constructive: Intermediate Value Theorem creates root approximation algorithm
- Non-constructive: Axiom of Choice proves existence without specific construction
- Constructive proofs in computer science link to Curry-Howard correspondence (proofs as programs)

Implications for mathematical practice

- Set theory impacted unrestricted comprehension principle rejected constructive set theories developed (CZF)
- Computational mathematics alignment proofs and algorithms more closely linked
- Computer-assisted proof systems enhanced by constructive approach
- Philosophical shift understanding of mathematical truth and existence challenged
- Platonist philosophy of mathematics questioned
- Reformulation of theorems and proofs often necessary
- Constructive alternatives to classical results developed
- Category theory and topos theory connections strengthened
- Theoretical computer science and type theory applications expanded

13.2 Set theory in topos

Set theory in topos generalizes traditional set theory to broader mathematical structures. It redefines sets as objects and functions as morphisms, allowing for more flexible reasoning and constructions within categorical frameworks.

Topos theory's internal logic uses intuitionistic principles, rejecting the law of excluded middle. This approach enables new perspectives on foundational concepts like equality, infinity, and power sets, expanding mathematical possibilities beyond classical set theory.

Foundations of Set Theory in Topoi

Concept and role of topos

- Topos definition encapsulates categories with finite limits, exponentials (function objects), and subobject classifier enabling generalization of set-theoretic concepts
- Objects in a topos function as generalized sets while morphisms act as generalized functions allowing for broader mathematical structures
- Key properties of a topos include cartesian closed category structure and existence of power objects facilitating advanced categorical constructions
- Examples of topos encompass category of sets, sheaves on a topological space, and presheaf categories demonstrating versatility in mathematical applications

Internal logic of topos

- Internal language of a topos utilizes higher-order intuitionistic logic and type theory interpretation for formal reasoning
- Kripke-Joyal semantics defines truth values as subobjects of terminal object and interprets logical connectives within topos framework
- Set-theoretic constructions in topos logic represent intersection as pullback, union as pushout, and complement using subobject classifier

- Reasoning techniques employ Mitchell-Bénabou language and geometric logic for rigorous proofs within topos theory

Set-Theoretic Structures in Topoi

Categorical constructions in topos

- Sets as objects conceptualize terminal object as singleton set and initial object as empty set providing foundational elements
- Functions as morphisms utilize identity morphism and composition of morphisms to represent mappings between objects
- Relations as subobjects employ monomorphisms and equalizers to capture relationships between objects
- Constructions using limits and colimits implement products for Cartesian products and coproducts for disjoint unions
- Exponential objects for function spaces leverage Curry-Howard correspondence connecting logic and computation
- Subobject classifier for power sets employs characteristic functions to define subsets within topos framework

Topos vs classical set theory

- Constructive logic in topos rejects law of excluded middle and adopts intuitionistic interpretation of negation
- Axiom of choice not generally valid in topos, though weaker forms may hold in specific contexts
- Powerset construction restricted in some topos, impacting size considerations in categorical set theory
- Equality and extensionality differentiate between intensional and extensional equality, emphasizing role of equivalence relations
- Foundation and regularity axioms allow for non-well-founded sets in certain topos, expanding set-theoretic possibilities
- Replacement and separation axioms find categorical analogues within topos framework
- Infinite sets represented through natural number object with associated induction principles for reasoning about infinite structures

13.3 Independence results and alternative foundations

Independence results in mathematics reveal the limitations of axiomatic systems and challenge the notion of absolute mathematical truth. They show that certain statements can't be proven or disproven within a given system, demonstrating the existence of multiple consistent models of mathematics.

Topos theory offers an alternative foundational framework, generalizing set-theoretic concepts to a category-theoretic setting. It provides a common language for different approaches, supports constructive mathematics, and enables the study of non-classical logics, offering new perspectives on mathematical foundations.

Independence Results in Mathematics

Significance of independence results

- Independence results reveal limitations of axiomatic systems showing certain statements cannot be proven or disproven within a given system and demonstrate existence of multiple consistent models of mathematics (Gödel's incompleteness theorems)
- Continuum hypothesis (CH) states no set with cardinality between integers and real numbers proven independent of ZFC (Zermelo-Fraenkel set theory with Axiom of Choice)
- Challenge notion of absolute mathematical truth and highlight role of axiom choice in mathematical foundations (Axiom of Choice, Banach-Tarski paradox)
- Motivate search for alternative foundational systems encouraging exploration of stronger axiom systems and inspiring development of new mathematical frameworks (large cardinal axioms, homotopy type theory)

Alternative Foundations and Topos Theory

Topos theory as foundational framework

- Generalizes set-theoretic concepts to category-theoretic setting providing common language for different foundational approaches (Grothendieck toposes, elementary toposes)
- Elementary topoi model intuitionistic logic supporting constructive mathematics and allowing interpretation of type theories (intuitionistic type theory, Martin-Löf type theory)
- Subobject classifier in topoi generalizes notion of truth values enabling study of non-classical logics (fuzzy logic, quantum logic)
- Internal language of a topos facilitates translation between categorical and logical concepts allowing formulation of mathematical theories within a topos (Mitchell-Bénabou language, Kripke-Joyal semantics)

Comparison of foundational approaches

- Classical set theory (ZFC)
- Strengths: well-established widely accepted powerful (Cantor's theorem, Zorn's lemma)
- Limitations: relies on law of excluded middle non-constructive (Banach-Tarski paradox, non-measurable sets)
- Constructive mathematics
- Strengths: computational content algorithmic interpretations (Bishop's constructive analysis, Brouwer's intuitionism)
- Limitations: rejection of certain classical theorems potential loss of expressive power (law of excluded middle, axiom of choice)
- Category-theoretic foundations
- Strengths: unifying framework abstract structural approach (ETCS, Lawvere's ETCC)

- Limitations: less intuitive for some mathematicians requires new conceptual framework (Yoneda lemma, adjoint functors)
- Type theories
- Strengths: close connection to computer science formal verification (Coq, Agda)
- Limitations: can be more complex may require additional axioms for classical results (dependent types, universe polymorphism)

Implications of alternative foundations

- Philosophical implications
- Ontological questions about mathematical objects (platonism, formalism)
- Nature of mathematical truth and certainty (Gödel's incompleteness theorems, independence results)
- Role of intuition and formalism in mathematics (intuitionism, logicism)
- Practical implications for research
- Choice of foundation may influence research directions (homotopy type theory, univalent foundations)
- Potential for new insights and connections between fields (derived algebraic geometry, higher category theory)
- Impact on proof techniques and methods (proof assistants, automated theorem proving)
- Educational considerations
- Challenges in teaching alternative foundations (category theory, type theory)
- Potential benefits for developing mathematical reasoning skills (constructive proofs, structural thinking)
- Preparation for interdisciplinary work (computer science, physics)
- Impact on mathematical practice
- Influence on problem-solving approaches (categorical methods, type-theoretic reasoning)
- Potential for new areas of mathematical investigation (higher-dimensional algebra, synthetic mathematics)
- Effects on collaboration and communication between mathematicians (formalization of mathematics, proof assistants)

Unit 14 – Topos Theory: Advanced Applications

14.1 Topoi in computer science and logic

Topoi bridge computer science, logic, and mathematics, offering powerful frameworks for understanding programming languages, type theory, and categorical logic. They provide a foundation for intuitionistic reasoning and constructive mathematics, challenging classical logic's assumptions.

In computer science, topoi illuminate programming language semantics and type theory. In logic, they offer alternative foundations for mathematics and support non-classical logics like modal, temporal, and quantum logic, expanding our tools for reasoning about complex systems.

Topoi in Computer Science and Logic

Applications of topoi in computer science

- Programming language semantics provide formal mathematical models for understanding and reasoning about programming languages
- Denotational semantics maps program constructs to mathematical objects in a domain
- Domain theory studies partially ordered sets used to model computation and define semantics
- Type theory formalizes mathematical foundations of programming languages and proof assistants
- Polymorphic lambda calculus extends simple typed lambda calculus with universal quantification over types
- Dependent types allow types to depend on values, enabling more expressive specifications
- Category theory in computer science applies categorical concepts to program design and analysis
- Functors as data structures represent container types (lists, trees) with associated operations
- Monads for computational effects encapsulate and compose side effects (state, I/O) in pure functional languages

Topoi and intuitionistic logic

- Constructive mathematics emphasizes computational content and constructive proofs
- Rejection of law of excluded middle avoids non-constructive existence proofs
- Proof-relevance treats proofs as first-class objects, not just truth values
- Heyting algebras generalize Boolean algebras, modeling intuitionistic propositional logic
- Internal logic of topoi provides semantics for intuitionistic higher-order logic
- Kripke-Joyal semantics interprets intuitionistic logic in terms of topological spaces
- Truth values as open sets capture notion of partial knowledge or information
- Realizability theory connects proofs with computations

- Computational interpretation of logic assigns algorithms to logical formulas and proofs

Topoi in categorical logic

- Higher-order logical systems extend first-order logic with quantification over functions and predicates
- Lambda calculus provides foundation for functional programming and proof theory
- Typed lambda calculi add type systems to lambda calculus, increasing expressiveness and safety
- Internal language of a topos formalizes intuitionistic set theory within categorical framework
- Mitchell-Bénabou language provides syntax for working with objects and morphisms in a topos
- Categorical semantics interprets logical systems in terms of categorical structures
- Functorial semantics models theories as functors between categories
- Lawvere theories represent algebraic theories as categories with products
- Topoi as generalized universes of sets provide foundation for mathematics alternative to ZFC
- Grothendieck topoi arise from sites in algebraic geometry, generalizing notion of space
- Elementary topoi axiomatize properties of category of sets, supporting internal logic

Topoi for non-classical logic semantics

- Modal logic extends classical logic with operators for necessity and possibility
- Kripke semantics interprets modal formulas using possible world structures
- Sheaf semantics generalizes Kripke semantics using topological spaces and sheaves
- Temporal logic reasons about time-dependent properties in computer science and AI
- Linear temporal logic models time as a linear sequence of states
- Branching time logic allows for multiple possible futures from each state
- Fuzzy logic extends classical logic to handle degrees of truth
- Fuzzy set theory generalizes classical set theory with partial membership
- Many-valued logics use truth values beyond just true and false
- Quantum logic formalizes reasoning about quantum mechanical systems
- Orthomodular lattices provide algebraic structure for quantum propositions
- Quantum set theory develops set-theoretic foundations compatible with quantum mechanics

14.2 Cohomology theories and topos theory

Topos theory unifies cohomology, providing a general setting to define and compare different theories. It connects étale cohomology to Galois cohomology and crystalline cohomology to de Rham cohomology, using topoi to bridge these concepts.

Sheaf cohomology in topoi calculates cohomology of sheaves, using derived categories and Grothendieck topologies. This framework extends to algebraic geometry, defining schemes and exploring various topologies, while also connecting to motivic cohomology and non-commutative geometry.

Topos Theory and Cohomology

Topos theory and cohomology relationships

- Topos theory unifies cohomology theories providing general setting for defining and studying cohomology allowing comparison between different cohomology theories
- Étale cohomology defined in terms of étale topology on schemes connects to Galois cohomology for fields applied in arithmetic geometry (Weil conjectures)
- Crystalline cohomology defined for schemes in characteristic p relates to de Rham cohomology in characteristic 0 uses crystalline site as a topos
- Topos theory compares cohomology theories through functorial properties and spectral sequences relating different theories (Leray spectral sequence)

Topoi in sheaf cohomology

- Sheaf cohomology in topoi defines sheaves on a site calculates cohomology of sheaves in a topos
- Derived categories construct derived category of sheaves define derived functors and their universal properties (derived pushforward)
- Grothendieck topologies and sites define and exemplify sheaves on a site (Zariski, étale, fppf)
- Cohomological descent relates Čech cohomology to sheaf cohomology uses hypercoverings in descent theory

Topos theory in algebraic geometry

- Grothendieck's work on schemes and topoi defines schemes as locally ringed spaces explores Zariski topology and étale topology
- Geometric morphisms between topoi relate to morphisms of schemes define pullback and pushforward functors
- Points of a topos correspond to geometric points of schemes classify points in various topoi (classifying topos)
- Coherent topoi define properties relate to coherent schemes (noetherian schemes)

Topoi for motivic cohomology

- Motivic cohomology defined using category of mixed motives relates to algebraic K-theory (Bloch-Lichtenbaum spectral sequence)
- Voevodsky's approach uses simplicial presheaves develops A_1 -homotopy theory
- ℓ -adic cohomology compares with étale cohomology applies to Weil conjectures (Riemann hypothesis for varieties over finite fields)
- De Rham-Witt cohomology relates to crystalline cohomology applies to p -adic Hodge theory (Fontaine's period rings)
- Topoi in non-commutative geometry follows Connes' approach to non-commutative spaces interprets cyclic cohomology topos-theoretically

14.3 Higher-dimensional and ∞ -topoi

Higher-dimensional topoi take ordinary topoi to new heights, incorporating higher categorical structures. They model complex algebraic and geometric relationships, providing a framework for studying advanced mathematical concepts.

∞ -topoi push this even further, allowing for infinite hierarchies of morphisms. They capture rich homotopical information and play a crucial role in unifying homotopy theory with category theory, opening doors to new mathematical frontiers.

Higher-Dimensional Topoi

Concept of higher-dimensional topoi

- Higher-dimensional topoi generalize ordinary topoi to higher categorical settings incorporating higher-dimensional categorical structures (2-categories, n -categories)
- n -topoi model $(n-1)$ -types in homotopy theory capturing higher-dimensional algebraic and geometric relationships
- Key features include internal logic allowing formal reasoning within the topos and geometric realization connecting abstract structures to concrete spaces
- Role in higher category theory provides framework for studying higher categorical structures (higher groupoids, higher stacks)

Theory of ∞ -topoi

- ∞ -topoi generalize ordinary topoi to infinite-dimensional settings based on ∞ -categories allowing infinite hierarchies of morphisms
- Fundamental properties include descent theory for gluing local data and sheaf condition in ∞ -categorical context enabling consistent information assembly
- Comparison with ordinary topoi reveals richer structure capturing homotopical information (weak equivalences, higher homotopies)
- Examples of ∞ -topoi include ∞ -category of spaces and ∞ -category of ∞ -groupoids representing fundamental objects of study

Connections and Applications

Topoi in homotopy theory

- Higher topos theory unifies concepts from homotopy theory and category theory providing framework for abstract homotopy theory (homotopy types, homotopy groups)
- Homotopy type theory connects to higher-dimensional topoi through univalent foundations program formalizing mathematics in type theory
- Model categories relate to higher-dimensional topoi through Quillen model structures enabling rigorous study of homotopy theory
- Homotopy limits and colimits interpret in higher-dimensional topoi as universal constructions preserving homotopical information

- Grothendieck ∞ -groupoids serve as fundamental objects in higher topos theory representing higher-dimensional analogues of sets

Applications of higher-dimensional topoi

- Derived algebraic geometry uses ∞ -topoi as foundation for studying derived schemes and stacks generalizing classical algebraic geometry
- Higher-dimensional moduli spaces employ moduli stacks as objects in ∞ -topoi enabling study of deformation theory in higher categorical settings
- Brave new algebra utilizes spectral schemes and structured ring spectra extending algebraic concepts to homotopical settings
- Higher categorical cohomology theories generalize cohomology theories in ∞ -topoi allowing refined invariants for geometric objects
- Applications in physics include higher gauge theory and topological quantum field theories modeling higher-dimensional phenomena