ACLT: Algebra, Categories, Logic in Topology
- Grothendieck's generalized topological spaces (toposes)

Steve Vickers
CS Theory Group
Birmingham

3. Classifying categories

Maths generated by a generic model

Outline of course

1. Sheaves: Continuous set-valued maps

2. Theories and models: Categorical approach to many-sorted first-order theories.

3. Classifying categories: Maths generated by a generic model

4. Toposes and geometric reasoning: How to "do generalized topology".

3. Classifying categories

Geometric theories may be incomplete
- not enough models in Set
- category of models in Set doesn't fully describe theory

Classifying category - e.g. Lawvere theory
= stuff freely generated by generic model
- there's a universal characterization of what this means

For finitary logics, can use universal algebra
- theory presents category (of appropriate kind) by generators and relations

For geometric logic, classifying topos is constructed by more ad hoc methods.

Presentation independence

Theory is really theory presentation
- start from its ingredients (sorts, functions, predicates, axioms)
- can generate much more (terms, formulae, valid sequents)
- different presentations can generate essentially the same stuff

up to isomorphism

e.g. can rebrand -

- sort lists as sorts with projection functions
- predicates as sorts with inclusion functions
- functions as predicates (the graphs) with axioms for functionhood
- axioms as sequents derivable from other axioms

Two questions
1. How can we describe the abstract essence of a theory
- independent of how it was presented?
2. Can we define maps (model transformers) from one theory to another
- in a way that doesn't depend on presentation?

Example

T_1 has - one sort σ
- a unary predicate P
- model M = object M(σ) equipped with designated subobject M(P)

T_2 has - two sorts σ, τ
- a unary function i: τ -> σ
- an axiom to force i to be monic

Exercise: How does axiom work?

$$i(x) =_\sigma i(y) \vdash_{x,y:\tau} x =_\tau y$$

- model N = two objects N(σ), N(τ) with monic N(i): N(τ) -> N(σ)

Same models, despite totally different signatures and axioms
Same homomorphisms too - categories of models are equivalent

Two questions
1. How can we describe the abstract essence of a theory
- independent of how it was presented?
2. Can we define maps (model transformers) from one theory to another
- in a way that doesn't depend on presentation?

Category of models?

No -

Relies on *completeness* - enough models

Geometric logic is incomplete

Two questions
1. How can we describe the abstract essence of a theory
- independent of how it was presented?
2. Can we define maps (model transformers) from one theory to another
- in a way that doesn't depend on presentation?

Classifying category

Everything derivable from presentation

Derived formulae and sequents

Derived sorts
Derived functions

Derivations apply to models in categories other than Set

Presentation independence: Propositional case
Lindenbaum algebras

Lindenbaum algebra for theory
= formulae modulo logical equivalence

Everything derivable from presentation

Derived formulae and sequents

~~Derived sorts~~
~~Derived functions~~

Derivations apply to models in ~~categories~~ other than ~~Set~~

posets

$\{0,1\}$

look for models in arbitrary poset A
with sufficient structure
- to interpret connectives
- and validate logical rules

Same process as before:

- Interpretation I interprets propositional symbols as elements of A
- I extends to arbitrary formulae using structure of A
- For a model, the sequents $\phi \vdash \psi$ must be satisfied:
- require $I(\phi) \leq I(\psi)$

Example: classical logic uses Boolean algebras

Connectives - conjunction, disjunction, negation
+ logical rules - including excluded middle
=> algebra of Boolean algebras

Given a signature (propositional symbols):
- formula = term in Boolean algebra formed from those symbols

Example: geometric logic uses *frames*

Connectives - finite conjunction, arbitrary disjunction
+ logical rules - not including excluded middle
=> algebra of Boolean algebras

hence a complete lattice

Frame = lattice with finite meets, arbitrary joins
+ meet distributes over all joins

Frame homomorphism preserves finite meets, all joins

Frames as objects are the same as complete Heyting algebras!

But frame homomorphisms only preserve the geometric structure
- not the Heyting arrow or negation

Universal property of Lindenbaum algebra L(T)

construction - formulae modulo equivalence

derived from signature
- using logical connectives

derived from sequents
- using logical rules

specification - has universal (generic) model

interpret each propositional symbol P as its equivalence class in L(T)

sequents all satisfied in L(T) - by construction
- so get a model in L(T)
- generic model

Universal property of Lindenbaum algebra L(T)

let A be a poset of appropriate kind e.g. Boolean algebra for classical logic

frame for geometric

let M be a model of T in A

Then there is a unique homomorphism M': L(T) -> A of appropriate kind
that transforms generic model to (specific model) M

Proof sketch
For each prop. symbol P, must have M'([P]) = I_M(P)
For homomorphism M', must have M'([ϕ]) = I_M(ϕ)
Well defined, because M a model
- hence if ϕ, ψ equivalent then I_M(ϕ) = I_M(ψ)
M' *is* a homomorphism, because equivalence is algebraic congruence.

Models of T in A are equivalent to homomorphisms L(T) -> A

Universal algebra

Universal property says:

L(T) is algebra presented by generators and relations

generators G = propositional symbols from signature
relations R = axioms from theory
- in equational form
- $\phi \vdash \psi$ becomes relation $\phi \vee \psi = \psi$ (i.e. $\phi \leq \psi$)

e.g. for classical logic (Boolean algebras)

$L(T) = BA\langle G | R \rangle$

Boolean algebra presented by generators G subject to relations R

Universal algebra - for geometric logic (frames)

L(T) = Fr<G|T>
- also written Ω[T]   see why later

Care needed to show it exists in geometric case

Formulae form a proper class
- because of unbounded infinitary disjunctions

"formulae modulo equivalence" is problematic

logical rules (idempotence, distributivity)
=> every formula equivalent to disjunction of finite conjunctions
=> get set of equivalence classes
=> can construct frame Fr<G|R> with correct universal property for Lindenbaum algebra

Presentations = propositional geometric theories

Algebra		Logic
generators	G	signature - propositional symbols
relations $\bigwedge_i a_i \leq \bigvee_j \bigwedge_k b_{jk}$	R	axioms $\bigwedge_i a_i \vdash \bigvee_j \bigwedge_k b_{jk}$
presentation	T = (G, R)	theory (signature, axioms)
frame presented	Ω[T] = Fr<G|R>	Lindenbaum algebra (formulae modulo equivalence)

connectives: finite conjunction, arbitrary disjunction

Universal property of Ω[T] = Fr<G|R>

For any frame A, and for any -

Algebra

Function f: G -> A
respecting the relations R

Logic

Model of T in A

there is a unique frame homomorphism f': Fr<G|R> -> A
that agrees with f on generators G

Easier to see when X = 1,
A = ΩX = P(X)
= {truth values}

Locales: write [T] for locale with Ω[T] = Fr<G|R>
For any locale X,
maps f: X -> [T] in bijection with
models of T in ΩX - models of T "at X"
Points of [T] = models of T

Sequents v. congruence

logic -- algebra

Congruence on algebraic terms
- represents logical equivalence of formulae

Logical equivalence $\phi \dashv\vdash \psi$

= both sequents can be derived $\phi \vdash \psi \qquad \psi \vdash \phi$
- from axioms of theory
  + rules of logic

Sequents correspond to partial order in algebra $\phi \leq \psi$

if (equivalently)

$\phi = \phi \wedge \psi$ or $\phi \vee \psi = \psi$ or $\top = \phi \to \psi$

Classifying categories = categorical Lindenbaum algebras

How to do the same for first-order theories?

Again, must understand models in categories other than Set

Look for category with generic model
- and appropriate universal property

more important than knowing
how to construct it

Structure needed in category (limits? colimits? images? etc.)
- depends on logic being used

For geometric logic use Grothendieck toposes

Models in categories (other than Set)

* sort σ — as set $I(\sigma)$ — carrier
  sort list $\vec{\sigma}$ — $I(\vec{\sigma})$ } product of carriers
  context $\vec{x}:\vec{\sigma}$ — $I(\vec{x})$ } $I(\sigma_1)\times\cdots\times I(\sigma_n)$
* function symbol $f:\vec{\sigma}\to\tau$ — function $I(f): I(\vec{\sigma})\to I(\tau)$
* predicate $P \hookrightarrow \vec{\sigma}$ — subset $I(P)\subseteq I(\vec{\sigma})$

Interpret:

sorts ... objects (the carriers)
sort lists ... products of carriers
function symbols, terms in context ... morphisms
predicate symbols, formulae in context ... subobjects (monics)

Use categorical structure to analyse what is needed to interpret logic

To interpret terms: need finite products

$$\begin{tikzcd} & I(\tau_j) & \\ I(\vec{\sigma}) \arrow[ur, "I(\vec{x}.t_j)"] \arrow[r, "\langle I(\vec{x}.t_j)\rangle_j"'] \arrow[rr, bend right, "I(\vec{x}.f(\vec{t}))"'] & I(\vec{\tau}) \arrow[u, "pr_j"'] \arrow[r, "I(f)"] & I(\upsilon) \end{tikzcd}$$

tupling uses
product property

To interpret formulae

Finite limits are enough to get
- predicate symbol applied to

$$
\begin{tikzcd}
 & & I(\tau_j) \\
I(\vec{\sigma}) \arrow[urr, "I(\vec{x}.t_j)"] \arrow[rr, "\langle I(\vec{x}.t_j)\rangle_j"'] & & I(\vec{\tau}) \arrow[u, "pr_j"'] \\
I(\vec{x}.P(\vec{E})) \arrow[u, "\subseteq"] \arrow[rr] & & I(P) \arrow[u, "\subseteq"']
\end{tikzcd}
$$

inverse image is pullback

- equations

equalizer

$$
\begin{tikzcd}
I(\vec{x}.\, t_1 =_\tau t_2) \arrow[r, "\subseteq"] & I(\vec{\sigma}) \arrow[r, shift left, "I(\vec{x}.t_1)"] \arrow[r, shift right, "I(\vec{x}.t_2)"'] & I(\tau)
\end{tikzcd}
$$

- conjunction

intersection is pullback

$$
\begin{tikzcd}
 & I(\vec{x}.\phi_1) \arrow[dr, "\subseteq"] & \\
I(\vec{x}.\phi_1 \wedge \phi_2) \arrow[ur, "\subseteq"] \arrow[dr, "\subseteq"'] & & I(\vec{\sigma}) \\
 & I(\vec{x}.\phi_2) \arrow[ur, "\subseteq"'] &
\end{tikzcd}
$$

In addition to finite limits

quotient = coequalizer

With coequalizers:
- can get image factorization
- as coequalizer of kernel pair
(pullback of morphism against itself)

$$\begin{tikzcd} & X/E \cong \text{image} \arrow[dr] & \\ E \arrow[r, shift left] \arrow[r, shift right] & X \arrow[u] \arrow[r, "f"] & Y \end{tikzcd}$$

E = equivalence relation for f. x E x' if f(x) = f(x')
E = kernel pair for f = pullback of f against itself

e.g.

$$\begin{tikzcd} I(\vec{x}, y.\phi) \arrow[r, hook] \arrow[d] & I(\vec{\sigma},\tau) = I(\vec{\sigma}) \times I(\tau) \arrow[d, "pr"] \\ I(\vec{x}.\exists y{:}\tau.\phi) \arrow[r, hook] & I(\vec{\sigma}) \end{tikzcd}$$

image in $I(\vec{\sigma})$

Get existential quantification

In addition to finite limits

With coequalizers and coproducts (hence colimits):
- can get disjunction

image factorization

$$
\begin{tikzcd}
& I(\vec{x}.\phi_1) \arrow[dl] \arrow[d] \arrow[dr] & \\
I(\vec{x}.\phi_1) \amalg I(\vec{x}.\phi_2) \arrow[r, dashed] \arrow[rr, bend right=60] & I(\vec{x}.\phi_1 \vee \phi_2) \arrow[r, hook] & I(\vec{\sigma}) \\
& I(\vec{x}.\phi_2) \arrow[ul] \arrow[u] \arrow[ur] &
\end{tikzcd}
$$

copairing

Coproducts go beyond logic
- logic only accesses subobjects of products
- but they're still useful

With infinite coproducts
- get infinite disjunctions

Summary

With finite limits, can interpret
- n-ary functions and predicates, and their applications to terms
- finite conjunctions, equations

With arbitrary colimits as well,
- arbitrary disjunctions
- existential quantification
- hence geometric logic

Also need the limits and colimits to be well behaved
- or logic doesn't work correctly
- e.g. to get distributivity

Category of sheaves over X is good for geometric theories

Grothendieck discovered more, toposes.

Grothendieck topos (Giraud's Theorem)

- each hom-set is small

- has finite limits

- has set-indexed coproducts
- for which coproduct injections are monic and disjoint
- and which are stable under pullback

- has image factorization
- and it is stable under pullback

- equivalence relations are kernel pairs

all set-indexed colimits

- has a "separating" set A of objects: for any f,g: X -> Y,
  If for all Z in A, and for all h: Z -> X, we have fh = gh, then f = g.

Appropriate functors between Grothendieck toposes

important (geometric) structure is finite limits, arbitrary colimits
- cf. finite meets, arbitrary joins for propositional case (topology)

appropriate functors preserve that structure

Also interested in natural transformations between them

Note

Grothendieck toposes have other non-geometric structure
- e.g. exponentials, subobject classifier, powerobjects

they are
elementary toposes

The extra structure is not preserved by these functors F

Other connectives - e.g. negation

Natural deduction rules for introduction and elimination imply:

$$\frac{\phi \vdash \neg\psi}{\overline{\phi \wedge \psi \vdash \bot}}$$

excluded middle is separate

$$\top \vdash \psi \vee \neg\psi$$

$\neg\psi$ is weakest ϕ such that $\phi \wedge \psi \vdash \bot$

Need more categorical structure to get this Heyting negation.

Category of sheaves has it
- but without excluded middle
- and it doesn't work fibrewise
- not geometric

Geometric morphism $f: \mathcal{F} \to \mathcal{E}$

= adjunction $\mathcal{F} \underset{f_*}{\overset{f^*}{\underset{\perp}{\leftrightarrows}}} \mathcal{E}$ such that

inverse image part f^* (left adjoint)
preserves finite limits

f* automatically preserves all colimits

Conversely: If F: E -> F preserves finite limits, arbitrary colimits
then it has a right adjoint G, and (F -| G) is geometric morphism

Geometric morphisms are equivalent to the appropriate functors
between Grothendieck toposes

but backwards!

Natural transformations between geometric morphisms

$f, g : \mathcal{E} \longrightarrow \mathcal{F}$

Natural transformation α from f to g

= (definition) natural transformation α* from f* to g*

- equivalent to natural transformation α_* *from* g_* *to* f_*

Universal property of classifying category L(T)

There is a generic model of T in L(T)

let A be a category of appropriate kind

let M be a model of T in A

Then there is a unique (up to isomorphism) functor M': L(T) -> A
that preserves the structure used by the logic
and transforms generic model to (specific model) M
- up to isomorphism

L(T) defined up to categorical equivalence

Same as for propositional theories
and Lindenbaum algebras,
but using categories instead of
posets.

More precisely: take care of homomorphisms

objects: geometric morphisms
morphisms: natural transformations

objects: models of T in E
morphisms: homomorphisms

$$\begin{array}{ccc} \mathbf{Top}(\mathcal{E}, \mathcal{S}[\mathbb{T}]) & \longrightarrow & \mathbf{Mod}_{\mathcal{E}}(\mathbb{T}) \\ f & \longmapsto & f^*(M_G) \end{array}$$

is equivalence of categories for every Grothendieck topos E

1. If M a model of T in E, then it's isomorphic to f*(M_G) for some f

2. If f, g: E -> S[T], and h: f*(M_G) -> g*(M_G) a homomorphism,
then there is unique α: f -> g such that h = α*(M_G)

is equivalence of categories for every Grothendieck topos E

1. If M a model of T in E, then it's isomorphic to f*(M_G) for some f

Second part says -

2. If f, g: E -> S[T], and h: f*(M_G) -> g*(M_G) a homomorphism, then there is unique α: f -> g such that h = α*(M_G)

the carrier morphisms in h lift to carrier morphisms for all derived sorts, giving natural transformation α

Works for positive logics, such as geometric logic
- and fragments such as algebraic logic (operators and equational laws)

For more general logics, e.g. classical or intuitionistic,
- must restrict morphisms on both sides
- e.g. restrict to isomorphisms

see lecture 2:
"Homomorphisms preserve *some* formulae"

Classifying category for finitary positive logics

Use universal algebra to present L(T)
- as "category of appropriate kind"
- using generators and relations got from theory T

Theory of "categories of appropriate kind" is not single-sorted algebraic
- two sorts, for objects and morphisms
- operations may be partial
- e.g. composition of morphisms, formation of pullbacks

It is usually cartesian (essentially algebraic)
- domains of definition for partial operations are defined by equations

Then presentation by generators and relations still works

For non-positive logics?
Can still work if careful with notion of homomorphism

Example: Lawvere theories

Suppose T single sorted algebraic (e.g. theory of monoids)
- one sort σ
- no predicate symbols, only functions (operators)
- equational axioms

$$\text{true} \vdash_{\vec{x}} t_1 = t_2$$

"category of appropriate kind" has finite products
- and we want functors that preserve finite products

classifying category has countably many objects
- for σ and its finite powers

abstract clone

morphisms for operators and all derived operations
- and tuple maps to powers of σ

L(T) is the Lawvere theory for T

Universal property of classifying topos S[T]

S[T] has generic model M_G

Let E be a Grothendieck topos with model M of T.

Then there is a unique (up to isomorphism) geometric morphism
  f: E -> S[T]
such that M $\cong$ f*(M_G)

S[T] is:

"geometric mathematics freely generated by the generic model M_G"

Whole of S[T] can be derived from ingredients of M_G, using finite limits and arbitrary colimits.

Constructing S[T]

Most important message:

Classifying topos can be constructed, with required universal property

How to do it?

1. Manipulate T into form of a site:
- signature is a category C
- interpretations are required to be "flat" functors from C
- axioms of form

$$\mathrm{true} \vdash_{y:Y} \bigvee_i \exists x:X_i .\ y = u_i(x)$$

  where u_i: X_i -> Y morphisms in C

2. Use presheaves over C to adjoin arbitrary colimits

and preserves finite limits already in C

3. Use sheaves (pasting condition) to make axioms hold

S[T] for propositional geometric T

Already have frame Ω[T] as poset form of Lindenbaum algebra

Theorem S[T] is equivalent to topos of sheaves over Ω[T]

Lecture 1:
For a topological space, defn of sheaf as presheaf with pasting depends only on frame of opens.
It works equally well for an arbitrary frame.

Lecture 2:
S[T] classifies theory of completely prime filters of Ω[T]

Theorem For two propositional geometric theories T, T':
Geometric morphisms S[T] -> S[T']
are equivalent to
frame homomorphisms Ω[T] <- Ω[T']

Further reading

Frames and their presentations
Johnstone - Stone Spaces
Vickers - Topology via Logic

Grothendieck toposes and classifying toposes
Mac Lane and Moerdijk
Johnstone - Elephant
Also Vickers "Locales and toposes as spaces" - reader's guide through the standard texts

Universal algebra for cartesian theories
Palmgren and Vickers