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

Steve Vickers
CS Theory Group
Birmingham

2. Theories and models

Categorical approach to many-sorted first-order theories.

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".

2. Theories and models

(First order, many sorted)

Theory = signature + axioms
Context = finite set of free variables
Axiom = sequent

Models in Set
- and in other categories

Homomorphisms between models

Geometric theories

Propositional geometric theory => topological space of models.

Generalize to predicate theories?

Oddities 1: Many-sorted

A set of sorts for all terms, including variables

Arities (of predicates, function symbols) say
- not just *how many* arguments,
- also what their sorts are
- also (function symbols) the sort of the result

Single-sorted = ordinary first order logic
- all terms have the same sort

No-sorted = propositional logic
- no variables or terms
- no function symbols (no sort for result)
- predicates have no arguments
- hence only possible predicates are propositional symbols

Oddities 2: Contexts

Don't assume overall countable stock of (sorted) free variables.

Instead, introduce context, finite set of sorted free variables, as needed.

Terms, formulae, entailments are *in context,*
- describing free variables allowed (and their sorts).

$(\vec{x}.\, t)$

$(\vec{x}.\, \phi)$

$\phi \vdash_{\vec{x}} \psi$

Empty carriers

Empty context allows correct treatment of empty carriers.

e.g.
$$\dfrac{\forall y\ \phi(y) \vdash_x \phi(x) \qquad \phi(x) \vdash_x \exists y\ \phi(y)}{\forall y\ \phi(y) \vdash_x \exists y\ \phi(y)} \quad \checkmark$$

Entailment holds for every interpretation of x
- vacuously true if carrier empty

Can't deduce $\forall y\ \phi(y) \vdash \exists y\ \phi(y)$ ✗

- false for empty carrier

Oddities 3: Sequents

Deal with logics lacking some connectives

e.g. geometric logic has conjunction, disjunction, but not implication
- correspond to intersection and union of open sets in topology

Two-level fomalization:
- formulae in context are built up using available connectives
- sequents in context express entailment

$$\overbrace{\phi \vdash_{\vec{x}} \psi}^{\text{sequent}} \quad \forall \vec{x}(\phi \to \psi)$$

formulae

context

Oddities 4: Infinitary connectives

In particular: infinitary disjunctions
- for arbitrary unions of opens

Unexpected consequences

Algebraic treatment (e.g. Lindenbaum algebras) more ad hoc
- see next talk

Can characterize some models, e.g. natural numbers, uniquely up to isomorphism.
- hence logic takes on aspects of type theory

Not possible with finitary first-order logic

Oddities 5: Incompleteness

No completeness in general
Two possible interpretations:

~~Inference rules not strong enough to get all semantic entailments~~

or

Not enough models to support all syntactic distinctions

Solution Look for models in categories other than Set
- then there are enough models
- but category of models in Set, or monad on Set, don't describe theory adequately
- need "categorical Lindenbaum algebras" (see next talk)

Many-sorted, first-order theory - in some given logic

Theory = signature + axioms

Signature has
- sorts
- predicates
- function symbols

$P \subseteq \vec{\sigma}$

$f : \vec{\sigma} \longrightarrow \tau$

Each predicate or function symbols has an arity specifying
- number of arguments (finite, possibly zero)
- the sort of each argument
- the sort of the result (for a function symbol)

A constant is a 0-ary function (no arguments)

$c : 1 \longrightarrow \tau$

or $c : \tau$

e.g. $f : \sigma_1, \sigma_2 \longrightarrow \tau$

binary function, arguments of sorts σ_1, σ_2, and result of sort τ.

$P \subseteq \sigma_1, \sigma_2$

binary predicate

Terms, formulae in context

Given a signature:

Context = finite list of variables, each assigned a sort

$\vec{x} : \vec{\sigma}$

Given a context:

$(\vec{x} . t)$

Terms built from free variables (from context) and function symbols in the usual way.

$(\vec{x} . \phi)$

Formulae built from:
- predicates applied to terms (of correct sort)
- equations between terms of same sort
- simpler formulae, using connectives and quantifiers as permitted by the logic

We always assume equality predicates

Bound variables are outside the context

Sequents, axioms, theories

Given a signature:

A sequent is a context, together with two formulae in that context.

A theory is a signature, together with a set of sequents
(the axioms for the theory)

Example

Be pragmatic about notation!

Monoids
- one sort, M
- two function symbols
  1: M
  _ _: M,M -> M
- no predicates
- axioms

unit
infix multiplication

$$T \vdash_{x:M} 1x =_M x \wedge x1 =_M x$$

unit laws

$$T \vdash_{x,y,z:M} x(yz) =_M (xy)z$$

associativity

Example

Monoid actions
- two sorts, M, A
- function symbols and axioms as for Monoids
- another function symbol
  _ _: A,M -> A
- two more axioms

$$\top \vdash_{x:A} x1 =_A x$$

$$\top \vdash_{x:A,\,y,z:M} x(yz) = (xy)z$$

Geometric theories

Formulae built using: $\underbrace{\top, \wedge,}_{\text{finite conjunctions}} \underbrace{\bot, \bigvee,}_{\text{arbitrary disjunctions}} =, \exists$

Propositional fragment (no sorts)

No-sorted = propositional logic
- no variables or terms
- no function symbols (no sort for result)
- predicates have no arguments
- hence only possible predicates are propositional symbols

Signature
= set of propositional symbols

Formulae built with finite conjunctions, arbitrary disjunctions
- relate to finite intersections, arbitrary unions of open sets

Interpreting a signature

Suppose we are given a signature

Each sort σ is interpreted as a set (its carrier)

A context is interpreted as the product of the carriers of the sorts of the free variables

$\sigma_1 \times \cdots \times \sigma_n$ where each x_i in $\vec{x}$ has sort σ_i

Interpreting a signature

To define interpretation I:
- specify interpretations of sorts, function symbols and predicates (*)
- other parts can then be derived

* 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 $I(f) : I(\vec{\sigma}) \longrightarrow I(\tau)$
* predicate $p \hookrightarrow \vec{\sigma}$	subset $I(p) \subseteq I(\vec{\sigma})$

Interpreting terms

- always in context

If term $t : \tau$ in context $\vec{x} : \vec{\sigma}$

$$I(\vec{x}.t) : I(\vec{\sigma}) \longrightarrow I(\tau)$$

Tuples of values to instantiate variables in context

Evaluate t on a tuple

Define by structural induction

(1) For free variable in the context: use projection

$$I(\vec{x}.x_i) : I(\vec{\sigma}) = I(\sigma_1) \times \cdots \times I(\sigma_n)$$

$$\begin{array}{c} I(\sigma_1) \times \cdots \times I(\sigma_n) \\ \big\downarrow pr_i \\ I(\sigma_i) \end{array}$$

Get i'th component of tuple

Interpreting terms

If term $t:\tau$ in context $\vec{x}:\vec{\sigma}$

$$I(\vec{x}.t) : I(\vec{\sigma}) \longrightarrow I(\tau)$$

Define by structural induction

(2) For function application

Suppose $f : \vec{\tau} \to \upsilon$

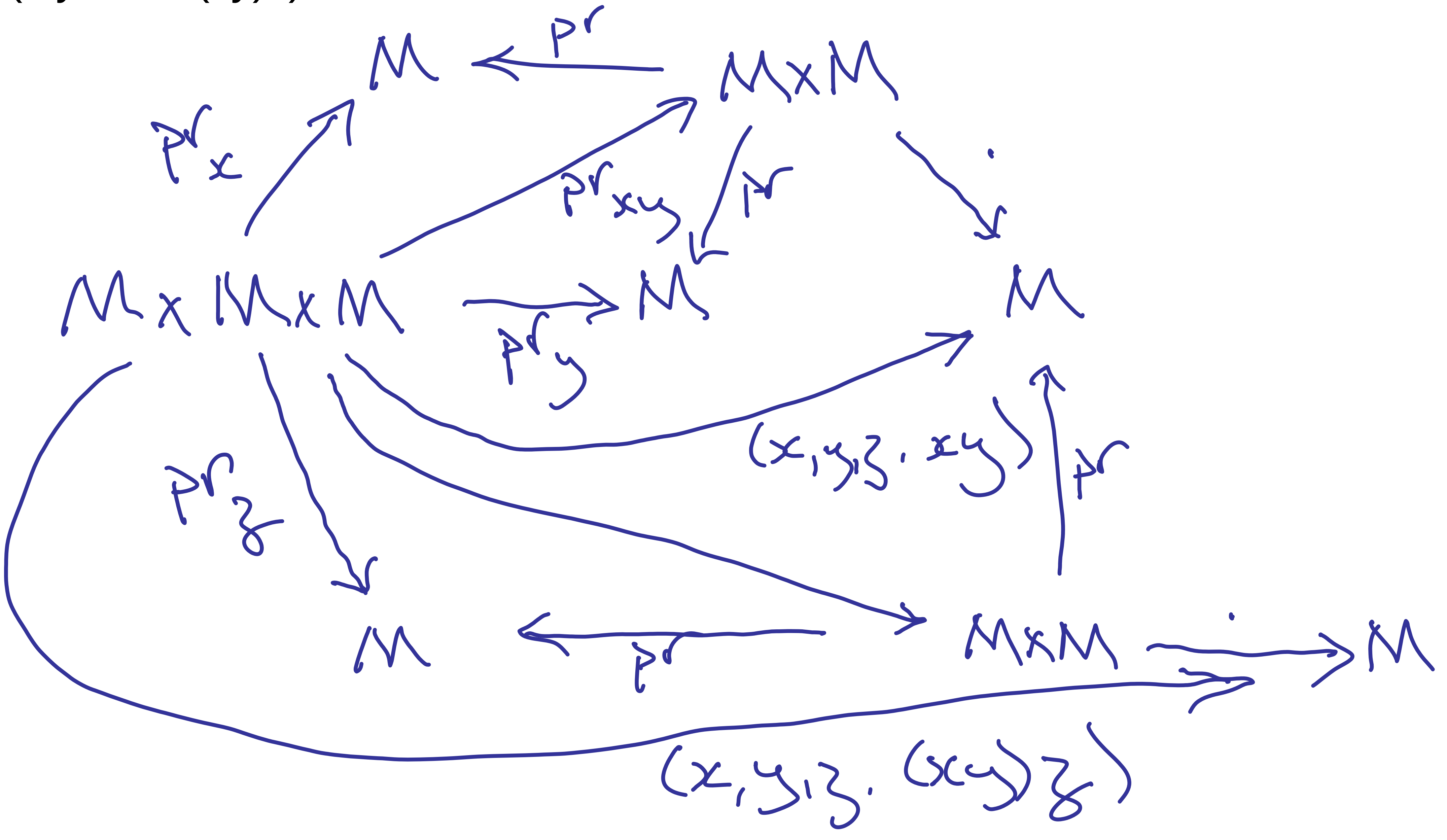
and $t_j : \tau_j$ in context $\vec{x}:\vec{\sigma}$

so $f(\vec{t}) : \upsilon$ in context $\vec{x}$

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

Example: Monoids

(x,y,z:M . (xy)z)



Interpreting formulae (in context)

If ϕ a formula in context $\vec{x} : \vec{\sigma}$

$$I(\vec{x}.\,\phi) \subseteq I(\vec{\sigma})$$

"the set of tuples for which ϕ holds"

Define by structural induction

(1) logical constants

$$I(\vec{x}.\,\top) = I(\vec{\sigma})$$ ~~~ Depends on context!

$$I(\vec{x}.\,\bot) = \emptyset$$

Interpreting formulae (in context)

If ϕ a formula in context $\vec{x} : \vec{\sigma}$
$\mathrm{I}(\vec{x} . \phi) \subseteq \mathrm{I}(\vec{\sigma})$

Define by structural induction

(1) predicate symbol applied to terms

Suppose $P \longrightarrow \vec{\tau}$

and $t_j : \tau_j$ in context $\vec{x} : \vec{\sigma}$

so $P(\vec{t})$ a formula in context $\vec{x}$

$$
\begin{tikzcd}
 & & \mathrm{I}(\tau_j) \\
\mathrm{I}(\vec{\sigma}) \arrow[urr, "\mathrm{I}(\vec{x}.t_j)"] \arrow[rr, "\langle \mathrm{I}(\vec{x}.t_j) \rangle_j"'] & & \mathrm{I}(\vec{\tau}) \arrow[u, "pr_j"'] \\
\mathrm{I}(\vec{x}.P(\vec{t})) \arrow[u, phantom, "\subseteq" rotate=90] \arrow[rr] & & \mathrm{I}(P) \arrow[u, phantom, "\subseteq" rotate=90]
\end{tikzcd}
$$

Inverse image of $\mathrm{I}(P)$:
set of tuples in $\mathrm{I}(\vec{\sigma})$
such that P holds for
their images under
the t_j's

Interpreting formulae (in context)

If ϕ a formula in context $\vec{x}:\vec{\sigma}$
$I(\vec{x}.\phi) \subseteq I(\vec{\sigma})$

Define by structural induction

(2) equation

terms $t_1, t_2 : \tau$ in context $\vec{x}:\vec{\sigma}$

formula $t_1 =_\tau t_2$

$$I(\vec{x}.\, t_1 =_\tau t_2) \subseteq I(\vec{\sigma}) \underset{I(\vec{x}.t_2)}{\overset{I(\vec{x}.t_1)}{\rightrightarrows}} I(\tau)$$

equalizer - those elements on whch two functions agree

Interpreting formulae (in context)

If ϕ a formula in context $\vec{x} : \vec{\sigma}$

$$I(\vec{x}.\phi) \subseteq I(\vec{\sigma})$$

Define by structural induction

(3) connectives - apply corresponding operations on subsets

ϕ_1, ϕ_2 formulae in context $\vec{x} : \vec{\sigma}$

e.g.

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

intersection

union

Interpreting formulae (in context)

If ϕ a formula in context $\vec{x}:\vec{\sigma}$

$I(\vec{x}.\phi) \subseteq I(\vec{\sigma})$

Define by structural induction

(4) quantifiers $\quad \phi$ formula in context $\vec{x}, y : \vec{\sigma}, \tau$

$\exists y:\tau.\phi \;,\; \forall y:\tau.\phi$ formulae in $\vec{x}:\vec{\sigma}$

e.g.

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

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

Models of a theory

An interpretation I satisfies a sequent $\phi \vdash_{\vec{x}:\vec{\sigma}} \psi$
if -

$$I(\vec{x}.\phi) \subseteq I(\vec{\sigma})$$

$$\cap$$

$$I(\vec{x}.\psi) \subseteq I(\vec{\sigma})$$

for every tuple: if ϕ holds then so does ψ

A model of a theory is
- an interpretation of its signature
- that satisfies all its axioms.

e.g. Monoids

Interpret signature:
set M with constant 1 and binary operation

Axioms: e.g. $\top \vdash_{x,y,z:M} x(yz) =_M (xy)z$

$$\begin{tikzcd} M(x,y,z.\,\top) \arrow[r, phantom, "\cong"] & M\times M\times M \arrow[r, shift left, "M(x,y,z.\,x(yz))"] \arrow[r, shift right, "M(x,y,z.\,(xy)z)"'] & M \\ & M(x,y,z.\,x(yz)=(xy)z) \arrow[u, phantom, sloped, "\subseteq"] & \end{tikzcd}$$

associativity holds for all triples of elements of M

Special case: propositional theories

No sorts to interpret

Empty context () interpreted as I() = nullary product 1

Propositional symbol P interpreted as subset I(P) of 1 = {*}
= truth value
- 1 is true, 0 (empty set) is false

Likewise, any formula ϕ
- connectives interpreted in lattice of truth values

Sequent $\phi \vdash \psi$ is satisfied means:
- if $* \in I(\phi)$ then $* \in I(\psi)$
- if $I(\phi)$ true then so is $I(\psi)$

Homomorphisms between models

M, N two models

homomorphism α: M -> N has

- for each sort σ, a carrier function α_σ: M(σ) -> N(σ)
- such that they preserve function symbols and predicates

$f : \vec{\sigma} \longrightarrow \tau$ $\qquad$ $p \longrightarrow \vec{\sigma}$

product of carrier functions

$$\begin{tikzcd} M(\vec{\sigma}) \arrow[r, "M(f)"] \arrow[d, "\alpha_{\vec{\sigma}}"'] & M(\tau) \arrow[d, "\alpha_\tau"] \\ N(\vec{\sigma}) \arrow[r, "N(f)"'] & N(\tau) \end{tikzcd} \qquad \begin{tikzcd} M(p) \arrow[r, phantom, "\subseteq"] \arrow[d, dotted] & M(\vec{\sigma}) \arrow[d, "\alpha_{\vec{\sigma}}"] \\ N(p) \arrow[r, phantom, "\subseteq"] & N(\vec{\sigma}) \end{tikzcd}$$

Models and homomorphisms form a category

Homomorphisms preserve all terms

$(\vec{x} : \vec{\sigma} .\ t : \tau)$

$$\begin{tikzcd}
M(\vec{\sigma}) \arrow[r, "M(\vec{x}.t)"] \arrow[d, "\alpha_{\vec{\sigma}}"'] & M(\tau) \arrow[d, "\alpha_{\tau}"] \\
N(\vec{\sigma}) \arrow[r, "N(\vec{x}.t)"'] & N(\tau)
\end{tikzcd}$$

By structural induction on t

Homomorphisms preserve some formulae

$(\vec{x}:\vec{\sigma}\,.\,\phi)$

Yes for geometric formulae
- use structural induction on formula

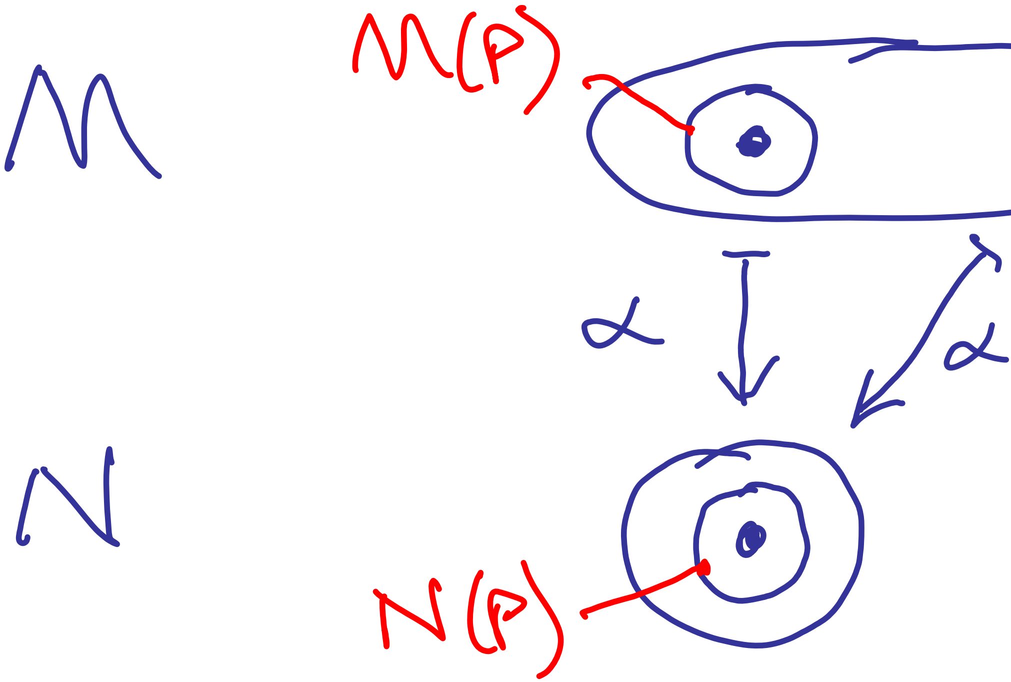
We shall be using homomorphisms for geometric theories

$$\begin{tikzcd} M(\vec{x}.\phi) \arrow[r, phantom, "\subseteq"] \arrow[d, dashed] & M(\vec{\sigma}) \arrow[d, "\alpha_{\vec{\sigma}}"] \\ N(\vec{x}.\phi) \arrow[r, phantom, "\subseteq"] & N(\vec{\sigma}) \end{tikzcd}$$

Homomorphisms preserve some formulae $(\vec{x} : \vec{\sigma} \,.\, \phi)$

No if formula uses negation, implication, or universal quantification

e.g. theory with one sort
and one unary predicate P
- model = set equipped with subset P
- homomorphism = function that restricts to the subsets



α doesn't preserve $(x . \neg P(x))$

Lecture 3: Important to lift model morphisms to other formulae. Hence classical logic needs different account of model morphism

Homomorphisms for propositional theories

No sorts
- no carrier functions required
- only one possible homomorphism
- but it only exists if all propositional symbols preserved
- models and homomorphisms form a poset

empty list of sorts

$$\begin{tikzcd} M(P) \arrow[r, leftrightarrow] \arrow[d, dotted] & 1 = M() \arrow[d, "\alpha_{()} = \text{unique function}"] \\ N(P) \arrow[r, leftrightarrow] & 1 = N() \end{tikzcd}$$

homomorphism M -> N
exists iff:
for every propositional
symbol P,
if P true for M
then it's also true for N

Geometric theories: Examples

1. Algebraic theories - e.g. monoids, monoid actions

Signature has sorts and function symbols
Formulae built with equations
Sequents all of form $\top \vdash_{\vec{x}} t_1 = t_2$

Geometric theories: Examples

2. Points of topological space X

Signature has
- no sorts (propositional theory)
- one propositional symbol P_U for each open U of X

Sequents

$$P_U \vdash P_V \quad \text{if } U \subseteq V$$

$$\top \vdash P_X$$

$$P_U \wedge P_V \vdash P_{U \cap V}$$

$$P_{\bigcup_i U_i} \vdash \bigvee_i P_{U_i}$$

converse sequents follow from first

infinite disjunctions!

Models?

Each P_U interpreted as subset of 1, i.e. truth value.

Let F = {U | P_U interpreted as true}

Axioms say -

- F is up-closed
- F contains X
- F is closed under intersection

$P_U \vdash P_V$ if $U \subseteq V$

$\top \vdash P_X$

$P_U \wedge P_V \vdash P_{U \cap V}$

$P_{\bigcup_i U_i} \vdash \bigvee_i P_{U_i}$

F is a filter in lattice O(X) of opens

- F "splits unions"

We say F is a completely prime filter

The models of the theory are the completely prime filters of O(X)

Neighbourhood filters

For each x in X:
- N_x = {open U | x in U}
- is a completely prime filter

$$N : X \longrightarrow \{\text{completely prime filters}\}$$

Note $x \sqsubseteq y$ iff $N_x \subseteq N_y$

y specializes x (x less than y in specialization order) if every open neighbourhood of x also contains y.

X is sober if N is a bijection

N is injective iff specialization is a partial order

Think: the completely prime filters are the true points
- if x, y have the same neighbourhood filter,
  they are just "different labels for the same point"

For sober spaces X, Y:

Maps f: X -> Y are in bijection with functions f*: Ω(Y) -> Ω(X) preserving finite intersections, arbitrary unions

Given f, f* is inverse image.

For reverse:
Completely prime filters of Ω(X) are equivalent to functions

Ω(X) -> Ω = P(1) = {truthvalues}

preserving finite intersections, arbitrary unions

Given f*, preserving those, by composition it transforms completely prime filters of Ω(X) to those of Ω(Y)

Hence by sobriety it gives f: X -> Y. It is continuous.

Point-free topology

Idea

Use a geometric theory to describe a topological space
Points = models of the theory
Opens = geometric formulae

Specifies points and opens all in one structure

This is point-free topology

Contrast with point-set topology
- first specify set of points
- then specify topology

Further reading

First order categorical logic
Johnstone - Elephant D1