1. Intro

2. Outline

3. Functorial semantics
   - Signatures to categories: objects
   - Signatures to categories: morphisms
   - The adjunction
   - Equational theories

4. Hyperdoctrines
   - Logic by adjointness

5. Mainstream approach
What is categorical logic?

- Many varieties of logics, here: mainly fragments of first-order logic, plus an incursion in (extensional) higher-order logic.
- Goal: uniform formulation of
  - their definitions,
  - the associated notions of models and maps between them.
- Tool: the informal notion of an internal language.
- Expository choice: lazy.
The three main approaches

- Allegories (Freyd): not covered here, but very effective.
- Variant: cartesian bicategories (Carboni and Walters).
Terms vs. formulas

First-order logic layers:
1. sorts, function symbols, equations,
2. first-order axioms.

For terms (and equations)

Categories with finite products, aka functorial semantics (Lawvere, 1963).

- With one sort $t$, terms $M(x_1, \ldots, x_n)$ are morphisms
  
  $$t \times \ldots \times t \rightarrow t$$

  in a category.

- Tuples represented by (formal) products $t \times \ldots \times t$. 
Hyperdoctrines

For formulas, naive idea:

- formulas are indexed over variables;
- $\leadsto$ hyperdoctrines, a kind of indexed categories.
Categories

A less naive idea:
- Start from terms, i.e., a category with finite products.
- Formulas add subobjects to terms:
  \[ \varphi(x) \hookrightarrow t. \]

Mainstream approach
All packed up into a category, logic done in terms of subobjects.
Allegories

A “converse” approach:

- instead of forcing formulas into terms,
- smoothly plunge operations into formulas:
  \[ f(x) \text{ viewed as a relation } y = f(x). \]
- Invent a calculus of relations: allegories.

Perhaps tighter:

- constructing an allegory from a theory is more direct,
- the logic is more primitive than with categories.

On the other hand: more \textit{ad hoc}.
Warm-up: Equational theories

**Definition**

An equational signature is

- a set $T$ of base types, or sorts,
- a set $F$ of function symbols, or operations, with arities $t_1, \ldots, t_n \rightarrow t$ in $T^{n+1}$.

Ex: magma, one sort $t$, one operation $t \cdot t \rightarrow t$: theory $\Sigma$.

- Any such signature $S = (T, F)$ freely generates a category with finite products.
- Let’s define this: categories, and then finite products.
Categories

Definition

A category $\mathcal{C}$ is

- a (possibly large) graph $\mathcal{C}_1 \xrightarrow{\text{s, t}} \mathcal{C}_0$,
- with an associative composition of edges
  $$ (A \xrightarrow{f} B \xrightarrow{g} C) \mapsto (A \xrightarrow{g \circ f} B), $$
- with units $A \xrightarrow{id_A} A$.

- Small: $\mathcal{C}_1$ and $\mathcal{C}_0$ are sets ($\neq$ classes).
- Locally small: for all $A, B \in \mathcal{C}_0$, $\mathcal{C}_1(A, B)$ is a set.
Boat examples of categories

- Locally small: sets, magmas, monoids, groups, . . .
- Small:
  - any preordered set,
  - the paths of any graph,
  - the homotopy classes of paths of any topological space.
Our example: the theory of magmas

- **Objects**: natural numbers,
- **Arrows** $p \to q$:
  - consider $p = \{0, \ldots, p - 1\}$ as a set of variables,
  - consider **terms** with variables in $p$, as generated by the grammar
    \[
    M, N, \ldots ::= x \mid M \cdot N, \quad x \in p,
    \]
    and call that $\mathcal{T}_\Sigma(p)$,
  - e.g., $(0 \cdot 0) \cdot 3 \in \mathcal{T}_\Sigma(4)$,
  - and let the set of arrows $p \to q$ be $\mathcal{T}_\Sigma(p)^q$.
- **Composition and identities?**
The theory of magmas

The composition $h$ of

$$p \xrightarrow{f} q \xrightarrow{g} r$$

is given by

$$h_k = g_k[f], \quad k \in r,$$

where

- $f$ is seen as the substitution $j \mapsto f_j$, for $j \in q$,
- and $g_k[f]$ replaces each $j$ with $f_j$ in $g_k$. 
The theory of magmas

Example of composition:

\[
3 \xrightarrow{(0\cdot 2),(2\cdot 1)} 2 \xrightarrow{0\cdot 1} 1
\]

compose to

\[
3 \xrightarrow{(0\cdot 2)(2\cdot 1)} 1.
\]

Less cryptic notation:

\[
x, y, z \xrightarrow{(x\cdot z),(z\cdot y)} u, v \xrightarrow{u\cdot v} 1
\]

compose to

\[
x, y, z \xrightarrow{(x\cdot z)(z\cdot y)} 1.
\]
Identities

- The identity on $p$ is the tuple $(0, \ldots, p - 1)$.
- Seen as a substitution, replaces $i$ with itself.
Summing up

- Objects: natural numbers,
- Arrows \( p \to q: \mathcal{T}_\Sigma(p)^q \).
- Composition by substitution.

Proposition

This yields a (small) category \( \mathcal{C}_\Sigma \).

Associativity is a variant of the standard substitution lemma

\[ f[g][h] = f[g[h]]. \]

Hence:

- The theory \( \Sigma \) of magmas yields a category \( \mathcal{C}_\Sigma \).
- Finite products?
Binary products

In any category $\mathcal{C}$, a product of two objects $A, B$ is:

- an object $C$
- arrows $\pi$ and $\pi'$

such that

$$A \leftarrow \pi \rightarrow C \rightarrow \pi' \rightarrow B$$
Binary products

In any category $\mathcal{C}$, a **product** of two objects $A, B$ is:

- an object $C$
- arrows $\pi$ and $\pi'$

such that for any $D, f, g$
Binary products

In any category $\mathcal{C}$, a **product** of two objects $A, B$ is:

- an object $C$
- arrows $\pi$ and $\pi'$

such that for any $D, f, g$, there is a unique arrow $h$ making both triangles commute.
Binary products

In any category $\mathcal{C}$, a **product** of two objects $A, B$ is:

- an object $C$
- arrows $\pi$ and $\pi'$

such that for any $D, f, g$, there is a unique arrow $h$ making both triangles commute.

Notation: $D = A \times B$, $h = \langle f, g \rangle$. 
Binary products

**Proposition**

*Binary products are unique up to unique commuting isomorphism.*

**Definition**

An *isomorphism* in a category $\mathcal{C}$ is an arrow $A \xrightarrow{f} B$ which has a two-sided inverse, i.e., a $g$ such that both

\[
\begin{align*}
A & \xrightarrow{f} B \\
\downarrow^{g} & \\
A & \xrightarrow{id_A} A
\end{align*}
\]

and

\[
\begin{align*}
B & \xrightarrow{g} A \\
\downarrow^{f} & \\
B & \xrightarrow{id_B} B
\end{align*}
\]

commute.
Binary products

Proposition

*Binary products are unique up to unique commuting isomorphism.*

For any two products

\[
\begin{array}{ccc}
A & & B \\
& C & \\
D & & \\
& & B
\end{array}
\]

\[\pi\quad \rho \quad \pi' \quad \rho'\]
Binary products

**Proposition**

*Binary products are unique up to unique commuting isomorphism.*

For any two products

```
A ← C → B
  |       |       |
  π      π'     ρ      ρ'
```

there is a unique isomorphism $i$ making both triangles commute.

- Proof actually quite subtle, let’s do it in detail.
First remark

Proposition

An iso has exactly one inverse.

Consider any two inverses $j$ and $j'$. The diagram

commutes, hence $j = j'$. 
Second remark

Given a commuting iso \( i \), i.e., one making

\[
\begin{array}{ccc}
\pi & \rightarrow & C \\
A & \downarrow & \pi' \\
\downarrow & i & \rightarrow \\
\rho & \rightarrow & B \\
\end{array}
\]

commute, its inverse \( j \) is also commuting. E.g., the diagram

\[
\begin{array}{ccc}
j & \rightarrow & C \\
D & \downarrow & \pi' \\
& i & \rightarrow \\
\id & \rightarrow & B \\
\end{array}
\]

commutes, hence \( \pi' \circ j = \rho' \). The rest symmetrically.
Third remark

\[ \langle \pi, \pi' \rangle = \text{id.} \] Indeed, \( \langle \pi, \pi' \rangle \) is the unique arrow making the diagram commute. But \( \text{id}_C \) does, hence \( \langle \pi, \pi' \rangle = \text{id.} \).
Proof of the proposition

Just for this proof, write:

- \langle f, g \rangle for product w.r.t. \( C, \pi, \pi' \),
- \([f, g]\) for product w.r.t. \( D, \rho, \rho' \).
Proof of the proposition: uniqueness

Any commuting inverses

meet the conditions for being respectively \([\pi, \pi']\) and \(\langle \rho, \rho' \rangle\). By uniqueness, they have to.
Proof of the proposition: existence

Construct

- The dashed composite meets the condition for being $\langle \pi, \pi' \rangle$, i.e., id, hence has to.
- By a symmetric argument, the dashed arrows are two-sided, commuting inverses.
Summary

We have proved:

**Proposition**

*Binary products are unique up to unique commuting isomorphism.*
\( \mathcal{C}_\Sigma \) has binary products

Slightly awkward: binary product in \( \mathcal{C}_\Sigma \) is actually \( \ldots \) sum:

\[
\begin{array}{c}
\downarrow \quad \downarrow \\
(0, \ldots, p - 1) & (p, \ldots, p + q - 1) \\
\leftarrow p & \to q,
\end{array}
\]

\( e.g., \)

\[
\begin{array}{c}
\downarrow \\
x \\
\leftarrow x \\
x, y, z (y, z) \\
\to y, z.
\end{array}
\]
Proof by example

E.g., the colored composite is

\[(y, z)[x \mapsto f_x, y \mapsto f_y, z \mapsto f_z],\]

i.e., \((f_y, f_z)\), as expected.
Proof by example

E.g., the colored composite is

$$(y, z)[x \mapsto f_x, y \mapsto f_y, z \mapsto f_z],$$

i.e., $(f_y, f_z)$, as expected.
Summary

We have (almost) proved:

Proposition

\[ C_\Sigma \text{ has binary products.} \]

More explicitly: any two objects have a binary product.
The nullary product

- From binary products, $n$-ary products.
- Associativity: $A \times (B \times C) \simeq (A \times B) \times C$.
- How about nullary product?
**The nullary product**

Mimicking the binary case with $2 \leadsto 0$:

- A nullary product for a 0-tuple of objects,
- is an object 1 (with void projections),
- such that for all object $D$ (and void arrows to the 0-tuple),
- there is a unique arrow $D \to 1$ (making the void diagram commute).
- But there is exactly one 0-tuple of objects.

**Compiling:**

- The nullary product, or **terminal object** is an object 1,
- such that for all object $A$, there is a unique arrow $A \to 1$.

**Ex:** in sets?
The terminal object in $\mathcal{C}_\Sigma$

**Proposition**

In $\mathcal{C}_\Sigma$, 0 is terminal.

Indeed, the unique morphism $A \to 0$ is the unique 0-tuple of terms in $\mathcal{T}_\Sigma(A)$. Hence:

**Proposition**

$\mathcal{C}_\Sigma$ has **finite products**, i.e., binary products and a terminal object.
Signatures to categories: morphisms

- We have constructed a function:
  \[ \mathcal{F}: \text{Sig} \rightarrow \text{FPCat} \]

- from signatures \( \Sigma = (T, F) \)
- to categories with finite products \( \mathcal{C}_\Sigma \).

But:
- There are natural morphisms of signatures
- and morphisms of categories with finite products.
- The assignment \( \mathcal{F} \) extends to morphisms, i.e., to a functor.
Consider any two categories $\mathcal{C}$ and $\mathcal{D}$.

**Definition**

A **functor** $F : \mathcal{C} \to \mathcal{D}$ is a morphism of graphs which preserves compositions and identities, i.e.,

$$F(g \circ f) = F(g) \circ F(f) \quad \text{and} \quad F(\text{id}_A) = \text{id}_{F(A)}$$

for all sensible $A, f, g$. 

*Hirschowitz Introduction to categorical logic 34/107*
The category of signatures

Let $\text{Sig}$ have

- objects: signatures $\Sigma = (T, F)$,
- arrows $\Sigma \rightarrow \Sigma'$ given by:
  - a function $f_0: T \rightarrow T'$, and
  - a function $f_1: F \rightarrow F'$ compatible with the arities.
Formalisation

As an exercise, let’s formalise this categorically.

**Definition**

For any set $X$, let $\mathcal{M}(X)$ be the free monoid on $X$, i.e., the set of finite words, or sequences on $X$.

$\mathcal{M}$ extends to a functor $\text{Set} \to \text{Set}$:

- Recall: Set is the category of sets and functions,
- For $f: X \to Y$, let

\[
\mathcal{M}(f): \quad \mathcal{M}(X) \quad \to \quad \mathcal{M}(Y) \\
(x_1, \ldots, x_n) \quad \mapsto \quad (f(x_1), \ldots, f(x_n)).
\]

It is actually a **monad**, as we’ll see later on.
Signatures as spans

- A signature \((T, F)\) is the same as a diagram
  \[
  \begin{array}{ccc}
  \mathcal{M}(T) & \xleftarrow{s} & F & \xrightarrow{t} & T.
  \end{array}
  \]

- An arrow \(\Sigma \to \Sigma'\) is a pair \((f_0, f_1)\) making
  \[
  \begin{array}{ccc}
  \mathcal{M}(T) & \xleftarrow{s} & F & \xrightarrow{t} & T \\
  \mathcal{M}(T') & \xleftarrow{s} & F' & \xrightarrow{t} & T' \\
  \end{array}
  \]
  commute.

- Composition of arrows is componentwise composition.

**Proposition**

*This yields a category \(\text{Sig}\) of signatures.*
The category of categories with finite products

What should a morphism $\mathcal{C} \to \mathcal{D}$ of categories with finite products be?

- A functor $\mathcal{C} \to \mathcal{D}$,
- preserving products, i.e.,

$$F(A \times B) = F(A) \times F(B) \quad \text{and} \quad F(1_{\mathcal{C}}) = 1_{\mathcal{D}}.$$

**Definition**

Call this a **finite product functor**.

**Details:**
- We assume finite products as chosen structure;
- Finite product functors preserve it strictly.
The category of categories with finite products

Proposition

The data:

- **objects**: (small) categories with finite products,
- **arrows**: finite product functors,
- **composition**: composition of finite product functors,

define a **(locally small)** category $\text{FPCat}$. 
The functor

The assignment

\[ F : \text{Sig} \to \text{FPCat} \]

- from signatures \( \Sigma = (T, F) \)
- to categories with finite products \( C_\Sigma \)

extends to a functor.
Let’s do that by example.
The functor by example

- Recall the theory Σ of magmas, with operation $t, t \to t$.
- Assume a morphism $f = (f_0, f_1): \Sigma \to \Sigma'$:

$$
\begin{array}{c}
\mathcal{M}(T) \leftarrow F = \{\cdot\} \xrightarrow{t} T = \{t\} \\
\mathcal{M}(f_0) \downarrow \quad f_1 \downarrow \quad f_0 \\
\mathcal{M}(T') \leftarrow F' \xrightarrow{t} T'.
\end{array}
$$

- In particular, let $t' = f_0(t)$.
- And let $*: t', t' \to t'$ be $f_1(\cdot)$.
The functor by example

Summary for $f$:

<table>
<thead>
<tr>
<th>Theory</th>
<th>Magmas $\Sigma$</th>
<th>$\Sigma'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic type</td>
<td>$t$</td>
<td>$t'$</td>
</tr>
<tr>
<td>Operation</td>
<td>$\cdot$</td>
<td>$\star$</td>
</tr>
</tbody>
</table>

Define $\mathcal{F}(f)$ to be the finite product functor $\mathcal{C}_\Sigma \to \mathcal{C}_{\Sigma'}$:

- on objects: $p \mapsto t'^p$, i.e., $t' \times \ldots \times t'$;
- on morphisms $p \xrightarrow{g} q$, define $\mathcal{F}(f)$
  - componentwise: $g = (g_1, \ldots, g_q)$,
  - and then by induction on terms:
    - $\mathcal{F}(f)(z) = z$,
    - $\mathcal{F}(f)(M \cdot N) = \mathcal{F}(f)(M) \star \mathcal{F}(f)(N)$. 
The functor $\mathcal{F}$, and back

We have constructed a functor

$$\mathcal{F} : \text{Sig} \to \text{FPCat},$$

sending $\Sigma$ to $\mathcal{F}(\Sigma) = C_{\Sigma}$ on objects.

Now, there is another functor

$$\mathcal{U} : \text{FPCat} \to \text{Sig},$$

sending any category with finite products $C$ to the signature with

- types the objects of $C$,
- operations $c_1, \ldots, c_n \to c$ all morphisms

$$c_1 \times \cdots \times c_n \to c.$$
The unit

The signature $\mathcal{U}(\mathcal{C})$ is big.

Example: for magmas, the signature $\mathcal{U}(\mathcal{F}(\Sigma))$ has

- types the natural numbers,
- and operations $p_1, \ldots, p_n \to p$ all morphisms $p_1 \times \ldots \times p_n \to p$ in $\mathcal{C}_\Sigma$, i.e.,
- all $p$-tuples of terms in $\mathcal{I}(p_1 + \ldots + p_n)$. 
The unit

Observation: there is a morphism

$$\eta_\Sigma : \Sigma \rightarrow \mathcal{U}(\mathcal{F}(\Sigma)),$$

sending each operation to itself, seen as a morphism in $\mathcal{F}(\Sigma)$.

Example

- The operation $t, t \rightarrow t$ is sent to
- the morphism $t \times t \rightarrow t$,
- seen as an operation in $\mathcal{U}(\mathcal{F}(\Sigma))$. 
For all sensible $F$, the diagram

\[
\begin{array}{ccc}
\Sigma & \xrightarrow{\eta \Sigma} & \mathcal{U}(\mathcal{F}(\Sigma)) \\
F \downarrow & & \downarrow \mathcal{U}(\mathcal{F}(F)) \\
\Sigma' & \xrightarrow{\eta \Sigma'} & \mathcal{U}(\mathcal{F}(\Sigma'))
\end{array}
\]

commutes.

**Proof.**

Easy check: $\mathcal{F}(F)$ is defined by induction from $F$. \qed
Interlude: natural transformations

Definition

A natural transformation \( \alpha \) is a family of arrows \( \alpha_c : F(c) \to G(c) \) making the diagram commute for all sensible \( f \).
We have proved:

**Proposition**

The units $\Sigma \to \mathcal{U}(\mathcal{F}(\Sigma))$ form a natural transformation.
Universal property of the unit

For any $F$, there is a unique $\overline{F}$ making the following triangle commute:

\[\begin{array}{ccc}
\Sigma & \xrightarrow{\eta \Sigma} & \mathcal{U}(\mathcal{F}(\Sigma)) \\
\downarrow^{F} & & \downarrow^{\mathcal{U}(\overline{F})} \\
\mathcal{U}(\mathcal{C}) & & \mathcal{U}(\mathcal{C}),
\end{array}\]

where $\mathcal{F}(\Sigma)$.

\[\begin{array}{cc}
\mathcal{F}(\Sigma) & \xrightarrow{\mathcal{F}} \\
\downarrow & \\
\mathcal{C} & \xrightarrow{\overline{F}}
\end{array}\]
Idea of the proof

- Finite products are expressive enough to encode term formation.
- Ex:
  - if $F : t \mapsto t'$ and $\cdot \mapsto \star$,
  - given $M, N \in \mathcal{F}_\Sigma(p)$,
  - translated to $[M]$ and $[N]$,
  - send $M \cdot N$ to $t' \circ [M] \circ [N] \to t' \times t' \to t'$.
- The constraint that $F$ be functorial and preserve finite products forces it to be that way.

Informally

- The pair $(\Sigma, F)$ is an **internal language** of $\mathcal{C}$.
- $\mathcal{U}(\mathcal{C})$ is the **internal language** of $\mathcal{C}$ (actually, the version with equations, see below).
Definition

An adjunction \( \mathcal{C} \dashv \mathcal{D} \) is a natural transformation

\[
\begin{array}{ccc}
F & \dashv & G \\
\downarrow & & \downarrow \\
\mathcal{C} & \cong & \mathcal{D},
\end{array}
\]

such that for all \( f \) there is a unique \( \bar{f} \) making the triangle commute

\[
\begin{array}{ccc}
c & \xrightarrow{\eta_c} & G(F(c)) \\
\downarrow f & & \downarrow G(\bar{f}) \\
G(d) & \xrightarrow{G(f)} & G(d),
\end{array}
\]

where

\[
\begin{array}{ccc}
F(c) & \xrightarrow{\bar{f}} & d.
\end{array}
\]
Adjunctions

Terminology

\[
\begin{tikzcd}
\text{In } & \mathcal{C} & \mathcal{D} & \text{Out}
\end{tikzcd}
\]

\[F \dashv G, \text{ } F \text{ is the left adjoint and } G \text{ is the right adjoint.}\]

Proposition

*Saying that } F \text{ has a right adjoint determines } G \text{ up to iso, and conversely.}*
Your first adjunction!

We have proved:

**Proposition**

There is an adjunction $\text{Sig} \dashv \text{FPCat}$.

- Discovered and studied by Lawvere under the name **functorial semantics**.
- Generalises usual semantics in Set: models may exist in any category with finite products.
- Nicely ties syntax and semantics together.
And the previous ones

Actually, you’ve already seen three adjunctions:

- between sets and monoids: \( \text{Set} \xleftarrow{\bot} \text{Mon} \),
- between the diagonal and binary product: \( \Delta \xleftarrow{\bot} \text{C} \times \text{C} \),
- between \( ! \) and the terminal object: \( \text{C} \xleftarrow{\bot} 1 \).
An original slogan of category theory was:

Adjoint functors are everywhere.

- Very abstract and scary.
- Very powerful:
  - limits (generalising finite products),
  - colimits (idem for coproducts),
  - free constructions (e.g., algebraic),
  - a few more well-known “types” of adjunctions,
  - maybe a lot more to be discovered.
A bit more on adjunctions

Equivalent definition (among others):

**Definition**

An adjunction \( \mathcal{C} \leftarrow \mathcal{D} \xrightarrow{\perp} \mathcal{D} \xrightarrow{\mathcal{G}} \mathcal{C} \xleftarrow{\mathcal{F}} \mathcal{D} \) is a natural isomorphism

\[
\mathcal{D}(F(c), d) \xrightarrow{\mathcal{C}(c, G(d))} \mathcal{C}(c, G(d))
\]

of functors \( \mathcal{C}^{\text{op}} \times \mathcal{D} \to \text{Set} \).
Equational theories

- Up to now, signatures: sorts and operations.
- Routine, but bureaucratic generalisation:

**Definition**

An equational theory \( \tau = (T, F, E) \) is

- a signature \( \Sigma = (T, F) \), plus
- a set of equations, i.e., elements of \( \prod_{p \in \mathbb{N}} T_{\Sigma}(p)^2 \).

- A morphism of equational theories is
  - a morphism \( f \) of signatures,
  - such that \( F(f) \) respects the equations.

**Proposition**

*This yields a category \( \mathbf{ETh} \) of equational theories.*
Extending the adjunction

Too briefly:

- $\mathcal{F}$ extends to equational theories by quotienting $\mathcal{C}_\Sigma$ by the equations.
- $\mathcal{U}$ refines into a functor $\mathbf{FPCat} \rightarrow \mathbf{ETH}$:
  
  - $\mathcal{U}(\mathcal{C})$ comes with a morphism
    
    $$\mathcal{T}(\mathcal{U}(\mathcal{C})) \xrightarrow{h_\mathcal{C}} \mathcal{U}(\mathcal{C})$$

    interpreting terms in $\mathcal{C}$;
  
  - take as equations all terms identified by this $h_\mathcal{C}$. 
Extending the adjunction

Proposition

This still yields an adjunction

This is both a soundness and completeness theorem:

- \( \mathcal{U}(\mathcal{F}(\tau)) \) contains everything derivable from \( \tau \).
- Existence of \( \overline{\mathcal{F}} \) is soundness: derivable implies true in all models.
- Completeness: \( \mathcal{F}(\tau) \) is the generic model where exactly what’s derivable is true.
Summary of functorial semantics

Equational theories specify categories with finite products.

Otherwise said:

The semantics for equational theories is in categories with finite products.
Remark

Completely eluded here: the importance of monads in the picture, both

- as a tool for presenting the framework (e.g., $\mathcal{T}$ is a monad),
- as an alternative semantics.
<table>
<thead>
<tr>
<th>1</th>
<th>Intro</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Outline</td>
</tr>
<tr>
<td>3</td>
<td>Functorial semantics</td>
</tr>
<tr>
<td></td>
<td>Signatures to categories: objects</td>
</tr>
<tr>
<td></td>
<td>Signatures to categories: morphisms</td>
</tr>
<tr>
<td></td>
<td>The adjunction</td>
</tr>
<tr>
<td></td>
<td>Equational theories</td>
</tr>
<tr>
<td>4</td>
<td>Hyperdoctrines</td>
</tr>
<tr>
<td></td>
<td>Logic by adjointness</td>
</tr>
<tr>
<td>5</td>
<td>Mainstream approach</td>
</tr>
</tbody>
</table>
Hyperdoctrines

Now, how do formulas enter the picture?

- Consider an equational theory $\tau = (T, F, E)$,
- plus a set of formulas $A$ on the generated terms.
Observation 1

- Formulas make sense in a context, i.e., an object of $\mathcal{C}_\tau$.
- Example: $0 = 1$ makes sense in 2.
- More readable: $x = y$ makes sense in $x, y$.
- Above each object, actually a partially ordered set (poset).
- So we have an indexed poset:

$$\mathcal{H}: \text{ob } \mathcal{C}_\tau \rightarrow \text{PoSet}.$$
Observation 2

- Morphisms in $\mathcal{C}_\tau$ act on formulas:

\[ \varphi(1, \ldots, q) \]

\[ p \xrightarrow{f} q \]

- Contravariantly.
- Functorially: $\varphi(f(g(x))) = \varphi(f \circ g(x))$, i.e.,

\[ \varphi \cdot f \cdot g = \varphi \cdot (f \circ g). \]

- So we have a functor:

\[ \mathcal{H}: \mathcal{C}_\tau^{op} \to \text{PoSet}. \]
Observation 2

- Morphisms in $\mathcal{C}_\tau$ act on formulas:

$$\varphi(f_1, \ldots, f_q) \leftarrow \varphi(1, \ldots, q)$$

- Contravariantly.
- Functorially: $\varphi(f(g(x))) = \varphi(f \circ g(x))$, i.e.,

$$\varphi \cdot f \cdot g = \varphi \cdot (f \circ g).$$

- So we have a functor:

$$\mathcal{H}: \mathcal{C}_\tau^{op} \rightarrow \text{PoSet}.$$
Generalisations

Two directions:

- proof theory: replace PoSet with Cat,
- replace 'functor' with 'pseudofunctor' (or 'fibration').

Not pursued here.
Logic in a hyperdoctrine

So:

- We interpreted equational theories $\tau$ in categories with finite products $C_\tau$.
- The proposal is to interpret logic over $\tau$ as a functor

$$\mathcal{H}: C_\tau^{op} \to \text{PoSet}.$$ 

What to interpret?

- Equational theories: substitution as composition.
- Logic: implication as ordering; other connectives?
Definition

A Heyting algebra is a which is bicartesian closed as a category. A morphism between such is a structure-preserving, monotone map.

I.e., we may interpret logic with $\top$, $\bot$, $\land$, $\lor$, $\Rightarrow$ in functors

$$\mathcal{C}_\mathcal{T}^{op} \to \text{HA},$$

where HA is the category of Heyting algebras.
Adjuncions

This defines connectives in terms of adjunctions

\[
\begin{align*}
\phi \leq \psi & \quad \phi \leq \theta \\
\phi \leq \psi \land \theta & \\
\phi \leq \theta & \quad \psi \leq \theta \\
\phi \lor \psi \leq \theta & \\
\phi \land \psi \leq \theta & \\
\psi \leq (\phi \Rightarrow \theta)
\end{align*}
\]
Quantifiers

- Define $\forall$ by adjunction:

$$\varphi(x) \leq \psi(x, y) \quad \frac{\varphi(x) \leq \psi(x, y)}{\varphi(x) \leq \forall y. \psi(x, y)}.$$

- But $\varphi(x)$ lives over the object $x$, while
- the inequality $\varphi(x) \leq \psi(x, y)$ really lives over $x, y$.
- So we should write

$$\left(\varphi \cdot \pi\right)(x, y) \leq \psi(x, y),$$

where

- $\pi : x, y \to x$ is the projection, and
- $(\varphi \cdot \pi)(x, y) = \mathcal{H}(\pi)(\varphi)(x, y) = \varphi(x)$.
Quantifiers
Rephrasing:

\[ \varphi \cdot \pi(x, y) \leq \psi(x, y) \]

\[ \varphi(x) \leq \forall y. \psi(x, y). \]

Hence:

Observation 1
Universal quantification is a map of posets

\[ \mathcal{H}(p + 1) \to \mathcal{H}(p) \]

right adjoint to \( \mathcal{H}(\pi) \).

Remark: not in HA, since in general

\[ \forall x. (\varphi \Rightarrow \psi) \neq (\forall x. \varphi) \Rightarrow (\forall x. \psi). \]
Quantification and substitution

- Assuming no variable capture, substitution interacts with $\forall$ via:
  \[
  (\forall x. \varphi)[f] = \forall x. (\varphi[f]).
  \]

- The square

\[
\begin{array}{ccc}
p + 1 & \xrightarrow{\pi} & p \\
\downarrow \quad f + 1 & \downarrow f & \downarrow \\
q + 1 & \xrightarrow{\pi} & q
\end{array}
\]

is a pullback.
Interlude: pullbacks

As with products, but over a fixed object $D$:

\[
\begin{array}{ccc}
A & \longrightarrow & B \\
\downarrow & & \downarrow \\
C & \longrightarrow & D.
\end{array}
\]
Interlude: pullbacks

As with products, but over a fixed object $D$:

\[
\begin{array}{ccc}
X & \rightarrow & A \\
\downarrow & & \downarrow \\
C & \rightarrow & D
\end{array}
\]
Interlude: pullbacks

As with products, but over a fixed object $D$:
Quantification and substitution

Rephrasing \((\forall x. \varphi)[f] = \forall x. (\varphi[f]):\)

**Observation 2**

For any pullback square as above, the square

\[
\begin{array}{ccc}
\mathcal{H}(p + 1) & \xrightarrow{\forall} & \mathcal{H}(p) \\
\downarrow & & \downarrow \\
- \cdot (f + 1) & & - \cdot f \\
\mathcal{H}(q + 1) & \xrightarrow{\forall} & \mathcal{H}(q)
\end{array}
\]

commutes in PoSet.
Generalised quantifiers

To get hyperdoctrines,

- require such right adjoints
  - not only to $\neg \cdot \pi$,
  - but to arbitrary $\neg \cdot f$,

satisfying a similar condition, called a Beck-Chevalley condition,

- and require also left adjoints to encode $\exists$. 
Hyperdoctrines

Definition

A (posetal, strict) hyperdoctrine is a functor $\mathcal{H}: C^{op} \rightarrow HA$ with left and right adjoints to all $\mathcal{H}(f)$:

$$\exists f \vdash (- \cdot f) \vdash \forall f$$

making

- for every pullback square as on the left
- the right-hand diagram commutes serially in PoSet:

\[
\begin{array}{ccc}
A & \xrightarrow{f} & C \\
\downarrow{h} & & \downarrow{k} \\
B & \xrightarrow{g} & D
\end{array}
\quad
\begin{array}{ccc}
\mathcal{H}(A) & \xrightarrow{\forall f, \exists f} & \mathcal{H}(C) \\
\downarrow{\mathcal{H}(h)} & & \downarrow{\mathcal{H}(k)} \\
\mathcal{H}(B) & \xrightarrow{\forall g, \exists g} & \mathcal{H}(D).
\end{array}
\]
Morphisms of hyperdoctrines

A morphism between hyperdoctrines $\mathcal{H}$ and $\mathcal{H}'$ is a diagram

\[
\begin{array}{ccc}
\mathcal{C}^{\text{op}} & \xrightarrow{F^{\text{op}}} & \mathcal{D}^{\text{op}} \\
\mathcal{H} & \xrightarrow{\alpha} & \mathcal{H}' \\
\text{HA,} & & \end{array}
\]

with

- $F$ preserving finite products, and
- $\alpha$ preserving $\forall$ and $\exists$. 
The category of hyperdoctrines

Proposition

*Hyperdoctrines and their morphisms form a category Hyp.*
First-order signatures

**Definition**

A 1st-order signature \( \Sigma = (T, F, R) \) consists of:

- \((T, F)\) is an equational signature,
- \(R\) is a set of relations, equipped with a function \( R \to \mathcal{M}(T) \).

Categorically, a diagram:

\[
R \xrightarrow{a} \mathcal{M}(T) \xleftarrow{s} F \xrightarrow{t} T.
\]

**Proposition**

*The obvious morphisms yield a category \( \text{Sig}_1 \).*
Theories

- Consider a signature $\Sigma = (T, F, R)$.
- Let $\text{Form}(\Sigma)$ be the set of formulas generated by $R, =, \land, \ldots$

**Definition**

A 1st-order *theory* $\tau$ consists of

- a 1st-order signature $\Sigma = (T, F, R)$, plus
- a set $E$ of *equations* in $\mathcal{T}(T, F)^2$, plus
- a set $A$ of *axioms* in $\text{Form}(\Sigma)$. 
Theories

A morphism between theories is
- a morphism between the underlying signatures,
- sending equations to equations,
- and axioms to axioms.

Proposition

This yields a category $\text{Th}_1$ of first-order theories.
The free hyperdoctrine

Given a theory $\tau = (T, F, R, E, A)$, construct a hyperdoctrine $\mathcal{H}_\tau$ with:

- base category $\mathcal{C}(T, F, E)$, terms modulo equations,
- over each object $c$, formulas in $(T, F, R)$ with variables in $c$, modulo provable equivalence.

Proposition

The assignment

$$\mathcal{F} : \text{Th}_1 \to \text{Hyp}$$

- from theories $\tau$
- to hyperdoctrines $\mathcal{H}_\tau$
extends to a functor.
The internal language of $\mathcal{H}$

**Definition**

Define $\mathcal{U}_0(\mathcal{H})$ to be the 1st-order signature with

- Types: the objects of $\mathcal{C}$.
- Operations $t_1, \ldots, t_n \rightarrow t$: morphisms $t_1 \times \ldots \times t_n \rightarrow t$.
- Relation symbols $R: t_1, \ldots, t_n \rightarrow prop$: objects of $\mathcal{H}(t_1 \times \ldots \times t_n)$.

How to deal with axioms?
Interpreting formulas

Interpret formulas over $\mathcal{U}_0(\mathcal{H})$ in context $\Gamma = (x_1 : t_1, \ldots, x_p : t_p)$ by induction:

$$
\varphi \quad \mapsto \quad \left[ \varphi \right]
$$

$$
R(f_1, \ldots, f_q) \quad \mapsto \quad R \cdot \langle f_1, \ldots, f_q \rangle
$$

where

$$
\begin{array}{c}
\Gamma \\
\xrightarrow{f} \\
\Delta
\end{array}
$$

Hirschowitz
Introduction to categorical logic
Interpreting formulas

Interpret formulas over $\mathcal{U}_0(\mathcal{H})$ in context $\Gamma = (x_1 : t_1, \ldots, x_p : t_p)$ by induction:

\[
\begin{align*}
\varphi & \mapsto [\varphi] \\
R(f_1, \ldots, f_q) & \mapsto R \cdot \langle f_1, \ldots, f_q \rangle \\
f = g & \mapsto Eq(\top) \cdot \langle f, g \rangle
\end{align*}
\]

where $f, g : \Gamma \to \Delta$. 

Hirschowitz
Introduction to categorical logic 84/107
Interpreting formulas

Interpret formulas over \( \mathcal{U}_0(\mathcal{H}) \) in context \( \Gamma = (x_1 : t_1, \ldots , x_p : t_p) \) by induction:

\[
\begin{align*}
\varphi & \mapsto \llbracket \varphi \rrbracket \\
R(f_1, \ldots , f_q) & \mapsto R \cdot \langle f_1, \ldots , f_q \rangle \\
f = g & \mapsto Eq(\top) \cdot \langle f, g \rangle \\
\top & \mapsto \top \\
\perp & \mapsto \perp \\
\varphi \land \psi & \mapsto \llbracket \varphi \rrbracket \land \llbracket \psi \rrbracket \\
\varphi \lor \psi & \mapsto \llbracket \varphi \rrbracket \lor \llbracket \psi \rrbracket \\
\varphi \Rightarrow \psi & \mapsto \llbracket \varphi \rrbracket \Rightarrow \llbracket \psi \rrbracket \\
\forall x : t. \varphi & \mapsto \forall \pi \llbracket \varphi \rrbracket \\
\exists x : t. \varphi & \mapsto \exists \pi \llbracket \varphi \rrbracket 
\end{align*}
\]

where \( \pi : \Gamma , t \to \Gamma \) is the projection.
The internal language of $\mathcal{H}$

**Definition**

Define $\mathcal{U}(\mathcal{H})$ to be the 1st-order theory with

- Types: the objects of $\mathcal{C}$.
- Operations $t_1, \ldots, t_n \to t$: morphisms $t_1 \times \ldots \times t_n \to t$.
- Equations: those validated by $\mathcal{C}$.
- Relation symbols $R: t_1, \ldots, t_n \to prop$: objects of $\mathcal{H}(t_1 \times \ldots \times t_n)$.
- Axioms the formulas $\varphi$ in $\text{Form}(\mathcal{H})$ such that $\top \leq [\varphi]$.

**Proposition**

*This assignment extends to a functor* $\mathcal{U}: \text{Hyp} \to \text{Th}_1$.
The adjunction, at last

**Theorem**

*These functors define an adjunction*

\[
\begin{array}{ccc}
\text{Th}_1 & \cong & \text{Hyp} \\
\mathcal{F} & \dashv & \mathcal{U} \\
\end{array}
\]

*between first-order theories and hyperdoctrines.*

- Soundness: any derivable sequent holds in any model.
- Completeness: \(\mathcal{F}(\tau)\) validates exactly the provable.
A remark: internal equality

- Morphisms in $C_T$ are sometimes equal.
- That is external equality.
- A notion of equality internal to the logic may be specified by adjunction.
A remark: internal equality

- Consider duplication

\[ \delta : x : A, y : B \xrightarrow{(x, x, y)} u : A, u' : A, v : B. \]

We may define \( Eq(\varphi)(u, u', v) = \varphi(u, v) \land (u = u') \) by adjunction:

\[
\frac{\varphi(x, y) \leq \psi(x, x, y)}{\varphi(u, v) \land u = u' \leq \psi(u, u', v)} \quad \frac{\varphi \leq \psi \cdot \delta}{Eq(\varphi) \leq \psi}
\]
A remark: internal equality

- Then define equality of \( f, g : A \to B \) as

\[
Eq(\top) \cdot \langle f, g \rangle,
\]

i.e., \( \top \land (f = g) \).

- This (bidirectional) rule is interderivable with more usual rules for equality.
An important construction of hyperdoctrines

- Start from a small category $\mathcal{C}$ with finite limits.
- Let $\mathcal{H}_C : \mathcal{C}^{\text{op}} \to \text{PoSet}$
  \[ c \mapsto \text{Sub}(c), \]
  where $\text{Sub}(c)$ is the set of equivalence classes of monics into $c$. 
Interlude: monic arrows

Definition

An arrow $f : c \to d$ in $\mathcal{C}$ is monic when for all $g, h$ as in
\[
\begin{array}{ccc}
e & \overset{g, h}{\longrightarrow} & c \\
\end{array}
\]
\[
\begin{array}{ccc}
f & \longrightarrow & d \\
\end{array}
\]
such that $fg = fh$, also $g = h$.

- In Set: injective.
- Generally written $c \hookrightarrow d$. 
Interlude: monic arrows

**Proposition**

For any commuting triangle

\[
\begin{array}{ccc}
\text{c} & \xrightarrow{f} & \text{d} \\
\downarrow{u} & & \downarrow{v} \\
\text{e} & & \\
\end{array}
\]

\(f\) is monic.
Proposition

For any commuting triangle

\[
\begin{array}{c}
a \\ g, h \\ \downarrow u \\
\end{array}
\quad \quad \quad \quad \quad \quad \quad \quad \quad 
\begin{array}{c}
c \\ f \\ \downarrow v \\
\end{array}
\quad \quad \quad \quad \quad \quad \quad \quad \quad 
\begin{array}{c}
d \\ \downarrow e, \\
\end{array}
\]

\(f\) is monic.

Proof: Assume \(g\) and \(h\) such that \(fg = fh\).

- Then also \(vfg = vfh\) by composition with \(v\), i.e.,
- \(ug = uh\).
- Hence \(g = h\) since \(u\) is monic.
While we are at it

The dual is:

**Definition**

An arrow $f : c \to d$ in $\mathcal{C}$ is **epic** when for all $g, h$ as in

$$
\begin{array}{c}
  c \\
  \downarrow f \\
  d \\
  \downarrow g, h \\
  e
\end{array}
$$

such that $gf = hf$, also $g = h$.

- In $\text{Set}$: surjective (trap: not in monoids).
- Generally written $c \twoheadrightarrow d$.
- Mnemonic: $f$ should cover $d$ to detect differences.
Constructing hyperdoctrines

An important construction of hyperdoctrines

- Start from a small category $\mathcal{C}$ with finite limits.
- Let $\mathcal{H}_C : \mathcal{C}^{\text{op}} \rightarrow \text{PoSet}$
  
  $c \mapsto \text{Sub}(c)$,

  where $\text{Sub}(c)$ is the set of equivalence classes of monics into $c$.

Important point, $\text{Sub}(c)$ is a poset:

- between $u$ and $v$, at most one arrow since $v$ monic,
- no cycle since we have quotiented under isomorphism.

What does $\mathcal{H}$ on morphisms?
Constructing hyperdoctrines

- Why is \( u \cdot f \) monic?
- What does \(- \cdot f\) do on morphisms?
- Why is \( \mathcal{H} \) functorial?
Constructing hyperdoctrines

Why is $u \cdot f$ monic?

What does $- \cdot f$ do on morphisms?

Why is $\mathcal{H}$ functorial?
\( \mathcal{H} \) is functorial

**Lemma (The pullback lemma)**

*In a diagram*

\[
\begin{array}{ccc}
a & \longrightarrow & b & \longrightarrow & c \\
\downarrow & & \downarrow & & \downarrow \\
x & \longrightarrow & y & \longrightarrow & z
\end{array}
\]

*the left-hand square is a pullback iff the outer rectangle is.*

Hence \( u \cdot f \cdot g = u \cdot fg \), i.e., functoriality of \( \mathcal{H} \).
### When is $\mathcal{H}_C$ a hyperdoctrine?

**Definition**

A category with finite limits is:

1. **regular** if it has stable images,
2. **coherent** if regular with stable unions,
3. **effective** if it has stable quotients of equivalence relations,
4. **positive** if coherent with disjoint finite coproducts,
5. **Heyting** if the pullback functors have right adjoints.

A category with all that is a **Heyting pretopos**.

- This yields enough to interpret 1st-order logic in $\mathcal{H}_C$.
- Examples: conjunction, $\exists_f$, implication.
Conjunction

Conjunction is just pullback, i.e., intersection:

\[
\begin{array}{c}
  a \cap b \\
  \downarrow \\
  a
\end{array} \Rightarrow \begin{array}{c}
  b \\
  \downarrow \\
  c.
\end{array}
\]

**Proof.**

The subobject \( a \cap b \) is a product in the poset \( \text{Sub}(c) \).
Images and $\exists_f$

**Definition**

A category has **images** when every morphism has an initial epi-mono factorisation.

- The epi in such a factorisation has to be a **cover**, i.e., the only subobjects through which it factors are isomorphisms.
- Requiring images to be stable under pullback $\equiv$ requiring covers to be.
Images and $\exists_f$

This allows interpreting $\exists_f$:

\[
\begin{array}{ccc}
  a \cap b & \rightarrow & b \\
  u \downarrow & & \downarrow \exists_f(u) \\
  a & \rightarrow & c.
\end{array}
\]

Proof.

There is an isomorphism $\text{Sub}(c)(\exists_f, \nu) \cong \text{Sub}(a)(u, \nu \cdot f)$, for any $\nu$.\qed
Implication

Using the right adjoint $\forall_u$:

\[
\begin{array}{c}
\begin{array}{c}
\forall_u(u \cap v) \\
a
\end{array} \\
\begin{array}{c}
\downarrow \\
\begin{array}{c}
u \\
u \cap v
\end{array}
\end{array}
\end{array} \quad \begin{array}{c}
b \\
\begin{array}{c}
\downarrow \\
v
\end{array}
\end{array} \quad \begin{array}{c}
\begin{array}{c}
x \\
\forall_u(u \cap v)
\end{array}
\end{array} \\
\begin{array}{c}
\downarrow \\
c, \\
\begin{array}{c}
u \\
u \cap v
\end{array}
\end{array}
\end{array}
\]

let $(u \Rightarrow v) = \forall_u(u \cap v)$.

Explanation:

- $(\forall_u(u \cap v))(x) = \forall y : a. (u(y) = x) \Rightarrow (u \cap v)(y)$.
- But there is either zero or one such $y$.
- If zero, then $x \not\in u$ and $(u \Rightarrow v)$ holds.
- If one, then $x \in u$ and $(u \Rightarrow v)(x) = (u \cap v)(x)$.
Ad hoc?

- The conditions above are modular, but somewhat \textit{ad hoc}.
- There is a particular case that implies them all, and more:

\begin{definition}

A \textbf{topos} is a category \( \mathcal{C} \)
- with finite limits,
- equipped with an object \( \Omega \)
- and a function \( P : \mathcal{C}_0 \rightarrow \mathcal{C}_0 \),
- with for each object \( c \) two isomorphisms

\[ Sub(c) \cong \mathcal{C}(c, \Omega) \hspace{1cm} \mathcal{C}(d \times c, \Omega) \cong \mathcal{C}(c, P(d)) \]

natural in \( A \).

Equivalent, elementary definition.
\end{definition}
Toposes

- The logic of $\mathcal{H}_C$ for a topos $\mathcal{C}$ is higher-order.
- Main examples of toposes: logic and sheaves.
- There is a characterisation of hyperdoctrines of the form $\mathcal{H}_C$.
- There is a slightly weaker notion of hyperdoctrines, triposes, which canonically generate toposes.
- They are important for Boolean- or Heyting-valued sets.
- They are important for realisability.

Let’s spare the definition for M. Hyland’s lecture.