# Introduction to Categorical Logic

[DRAFT: January 22, 2023]

Steve Awodey Andrej Bauer

# Contents

# Chapter 1

# Algebraic Theories

Algebraic theories are descriptions of structures that are given entirely in terms of operations and equations. All such algebraic notions have in common some quite deep and general properties, from the existence of free algebras to Lawvere's duality theory. The most basic of these are presented in this chapter. The development also serves as a first example and template for the general scheme of *functorial semantics*, to be applied to other logical notions in later chapters.

## 1.1 Syntax and semantics

We begin with a general approach to algebraic structures such as groups, rings, and lattices. These are characterized by axiomatizations which involve only a single sort of variables and constants, operations, and equations. It is important that the operations are defined everywhere, which excludes two important examples: fields, because the inverse of 0 is undefined, and categories because composition is defined only for certain pairs of morphisms.

Let us start with the quintessential algebraic theory: the theory of groups. In first-order logic, a group can be described as a set $G$ with a binary operation $\cdot : G \times G \to G$, satisfying the two first-order axioms:

$$\forall\, x, y, z \in G \,.\, (x \cdot y) \cdot z = x \cdot (y \cdot z)$$
$$\exists\, e \in G \,.\, \forall\, x \in G \,.\, \exists\, y \in G \,.\, (e \cdot x = x \cdot e = x \wedge x \cdot y = y \cdot x = e)$$

Taking a closer look at the logical form of these axioms, we see that the second one, which expresses the existence of a unit and inverse elements, is somewhat unsatisfactory because it involves nested quantifiers. Not only does this complicate the interpretation, but it is not really necessary, since the unit element and inverse operation in a group are uniquely determined. Thus we can add them to the structure and reformulate as follows. The unit is to be represented by a distinguished *constant* $e \in G$, and the inverse is to be a unary *operation* $^{-1} : G \to G$. We then obtain an equivalent formulation in which all axioms can be expressed as (universally quantified) *equations*:

$$\begin{gathered} x \cdot (y \cdot z) = (x \cdot y) \cdot z \\ x \cdot e = x \qquad e \cdot x = x \\ x \cdot x^{-1} = e \qquad x^{-1} \cdot x = e \end{gathered}$$

The universal quantifiers $\forall\, x \in G, \forall\, y \in G$, etc. are no longer needed in stating the axioms, since we can interpret all variables as ranging over all elements of $G$ (because of our restriction to totally defined operations). Nor do we really need to explicitly mention the particular set $G$ in the specification. Finally, since the constant $e$ can be regarded as a nullary operation, i.e., a function $e : \mathbf{1} \to G$, the specification of the group concept consists solely of operations and equations. This leads to the following general definition of an algebraic theory.

**Definition 1.1.1.** A *signature* $\Sigma$ for an algebraic theory consists of a family of sets $\{\Sigma_k\}_{k\in\mathbb{N}}$. The elements of $\Sigma_k$ are called the *$k$-ary operations.* In particular, the elements of $\Sigma_0$ are the *nullary operations* or *constants.*

The *terms* of a signature $\Sigma$ are the expressions constructed inductively by the following rules:

1. variables $x$, $y$, $z$, ..., are terms,
2. if $t_1, \ldots, t_k$ are terms and $f \in \Sigma_k$ is a $k$-ary operation then $f(t_1, \ldots, t_k)$ is a term.

**Definition 1.1.2** (cf. Definition **??**)**.** An *algebraic theory* $\mathbb{T} = (\Sigma_{\mathbb{T}}, A_{\mathbb{T}})$ is given by a signature $\Sigma_{\mathbb{T}}$ and a set $A_{\mathbb{T}}$ of *axioms*, which are equations between terms (formally, pairs of terms).

Algebraic theories are also called *equational theories.* We do not assume that the sets $\Sigma_k$ or $A_{\mathbb{T}}$ are finite, but the individual terms and equations always involve only finitely many variables.

**Example 1.1.3.** The theory of a commutative ring with unit is an algebraic theory. There are two nullary operations (constants) 0 and 1, a unary operation $-$, and two binary operations $+$ and $\cdot$. The equations are:

$$\begin{aligned} (x+y)+z &= x+(y+z) & (x\cdot y)\cdot z &= x\cdot(y\cdot z) \\ x+0 &= x & x\cdot 1 &= x \\ 0+x &= x & 1\cdot x &= x \\ x+(-x) &= 0 & (x+y)\cdot z &= x\cdot z+y\cdot z \\ (-x)+x &= 0 & z\cdot(x+y) &= z\cdot x+z\cdot y \\ x+y &= y+x & x\cdot y &= y\cdot x \end{aligned}$$

**Example 1.1.4.** The "empty" or trivial theory $\mathbb{T}_0$ with no operations and no equations is the theory of a set.

**Example 1.1.5.** The theory with one constant and no equations is the theory of a *pointed set*, cf. Example **??**.

**Example 1.1.6.** Let $R$ be a ring. There is an algebraic theory of left $R$-modules. It has one constant $0$, a unary operation $-$, a binary operation $+$, and for each $a \in R$ a unary operation $\overline{a}$, called *scalar multiplication by $a$*. The following equations hold:

$$\begin{aligned} (x+y)+z &= x+(y+z)\ , & x+y &= y+x\ , \\ x+0 &= x\ , & 0+x &= x\ , \\ x+(-x) &= 0\ , & (-x)+x &= 0\ . \end{aligned}$$

For every $a, b \in R$ we also have the equations

$$\overline{a}(x+y) = \overline{a}\,x + \overline{a}\,y\ , \qquad \overline{a}(\overline{b}\,x) = \overline{(ab)}\,x\ , \qquad \overline{(a+b)}\,x = \overline{a}\,x + \overline{b}\,x\ .$$

Scalar multiplication by $a$ is usually written as $a \cdot x$ instead of $\overline{a}\,x$. If we replace the ring $R$ by a field $\mathbb{F}$ we obtain the algebraic theory of a vector space over $\mathbb{F}$ (even though the theory of fields is not algebraic!).

**Example 1.1.7.** In computer science, inductive datatypes are examples of algebraic theories. For example, the datatype of binary trees with leaves labeled by integers might be defined as follows in a programming language:

```
type tree = Leaf of int | Node of tree * tree
```

This corresponds to the algebraic theory with a constant `Leaf` $n$ for each integer $n$ and a binary operation `Node`. There are no equations. Actually, when computer scientists define a datatype like this, they have in mind a particular model of the theory, namely the *free* one.

**Example 1.1.8.** An obvious non-example is the theory of posets, formulated with a binary relation symbol $x \leq y$ and the usual axioms of reflexivity, transitivity and anti-symmetry, namely:

$$\begin{gathered} x \leq x \\ x \leq y\ ,\ y \leq z \Rightarrow x \leq z \\ x \leq y\ ,\ y \leq x \Rightarrow x = x \end{gathered}$$

On the other hand, using an operation of greatest lower bound or "meet" $x \wedge y$, one can make the equational theory of "$\wedge$-semilattices":

$$\begin{gathered} x \wedge x = x \\ x \wedge y = y \wedge x \\ x \wedge (y \wedge z) = (x \wedge y) \wedge z \end{gathered}$$

Then, defining a partial ordering by $x \leq y \iff x \wedge y = x$ we arrive at the notion of a "poset with meets", which *is* equational (of course, the same can be done with joins $x \vee y$ as well). We will show later (in section **??**) that there is no reformulation of the general theory of posets into an equivalent equational one by considering the *category of models* of the theory, i.e. the category of posets, and showing that it lacks a general property enjoyed by all categories of algebras.

**Exercise 1.1.9.** Let $G$ be a group. Formulate the notion of a (left) $G$-set (i.e. a functor $G \to \mathsf{Set}$) as an algebraic theory.

### 1.1.1 Models of algebraic theories

Let us now consider *models* of an algebraic theory, i.e. *algebras.* Classically, a group can be given by a set $G$, an element $e \in G$, a function $m : G \times G \to G$ and a function $i : G \to G$, satisfying the group axioms:

$$\begin{aligned} m\left(x, m\left(y, z\right)\right) &= m\left(m\left(x, y\right), z\right) \\ m\left(x, i\,x\right) &= m\left(i\,x, x\right) = e \\ m\left(x, e\right) &= m\left(e, x\right) = x \end{aligned}$$

for any $x, y, z \in G$. Observe, however, that this notion can easily be generalized so that we can speak of models of group theory in categories other than $\mathsf{Set}$. This is accomplished simply by translating the equations between arbitrary elements into equations between the operations themselves: thus a group is given, first, by an object $G \in \mathsf{Set}$ and three morphisms

$$e : 1 \to G\,, \qquad m : G \times G \to G\,, \qquad i : G \to G\,.$$

The associativity axiom is then expressed by the commutativity of the following diagram:

$$\begin{tikzcd} G \times G \times G \arrow[r, "m \times \pi_2"] \arrow[d, "\pi_0 \times m"'] & G \times G \arrow[d, "m"] \\ G \times G \arrow[r, "m"'] & G \end{tikzcd} \tag{1.1}$$

Note that we have omitted the canonical associativity function $G \times (G \times G) \cong (G \times G) \times G$, which should be inserted into the top left corner of the diagram. The equations for the unit and the inverse are similarly expressed by commutativity of the following diagrams:

$$\begin{tikzcd}
G\times 1 \arrow[r, "1_G\times e"] \arrow[dr, "\pi_0"'] & G\times G \arrow[d, "m"] & 1\times G \arrow[l, "e\times 1_G"'] \arrow[dl, "\pi_1"] \\
& G &
\end{tikzcd}
\qquad
\begin{tikzcd}
G \arrow[r, "\langle 1_G, i\rangle"] \arrow[d, "!_G"'] & G\times G \arrow[d, "m"] & G \arrow[l, "\langle i, 1_G\rangle"'] \arrow[d, "!_G"] \\
1 \arrow[r, "e"'] & G & 1 \arrow[l, "e"]
\end{tikzcd} \tag{1.2}$$

This formulation makes sense in any category $\mathcal{C}$ with finite products.

**Definition 1.1.10.** Let $\mathcal{C}$ be a category with finite products. A *group in* $\mathcal{C}$ consists of an object $G$ equipped with arrows:

$$\begin{tikzcd}
G\times G \arrow[r, "m"] & G & G \arrow[l, "i"'] \\
& 1 \arrow[u, "e"'] &
\end{tikzcd}$$

such that the above diagrams (1.1) and (1.2) expressing the group equations commute.

There is also an obvious corresponding generalization of a group homomorphism in $\mathsf{Set}$ to homomorphisms of groups in $\mathcal{C}$. Namely, an arrow in $\mathcal{C}$ between (the underlying objects of) groups, say $h : M \to N$, is a homomorphism if it commutes with the interpretations of the basic operations $m$, $i$, and $e$,

$$h \circ m^M = m^N \circ h^2 \qquad h \circ i^M = i^N \circ h \qquad h \circ e^M = e^N$$

as indicated in:

$$\begin{tikzcd}
M^2 \arrow[r, "h^2"] \arrow[d, "m^M"'] & N^2 \arrow[d, "m^N"] \\
M \arrow[r, "h"'] & N
\end{tikzcd}
\qquad
\begin{tikzcd}
M \arrow[r, "h"] \arrow[d, "i^M"'] & N \arrow[d, "i^N"] \\
M \arrow[r, "h"'] & N
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[r, "="] \arrow[d, "e^M"'] & 1 \arrow[d, "e^N"] \\
M \arrow[r, "h"'] & N
\end{tikzcd}$$

Together with the evident composition and identity arrows inherited from $\mathcal{C}$, this gives a category of groups in $\mathcal{C}$, which we denote:

$$\mathsf{Group}(\mathcal{C})$$

In general, we define an *interpretation* $I$ of a theory $\mathbb{T}$ in a category $\mathcal{C}$ with finite products to consist of an object $I \in \mathcal{C}$ and, for each basic operation $f$ of arity $k$, a morphism $f^I : I^k \to I$. (More formally, $I$ is the tuple consisting of an underlying object $|I|$ and the interpretations $f^I$, but we shall write simply $I$ for $|I|$.) In particular, basic constants are interpreted as morphisms $1 \to I$. The interpretation is then extended to all terms as follows: a general term $t$ will be interpreted together with a *context of variables* $x_1, \ldots, x_n$ (a list without repetitions), where the variables appearing in $t$ are among those appearing in the context. We write

$$x_1, \ldots, x_n \mid t \tag{1.3}$$

for a term $t$ in context $x_1, \ldots x_n$. The interpretation of such a term in context (1.3) is a morphism $t^I : I^n \to I$, determined by the following specification:

1. The interpretation of a variable $x_i$ among the $x_1, \ldots x_n$ is the $i$-th projection $\pi_i : I^n \to I$.

2. A term of the form $f\,(t_1, \ldots, t_k)$ is interpreted as the composite:

$$\begin{tikzcd} I^n \arrow[r, "\left({t_1}^I, \ldots, {t_k}^I\right)"] & I^k \arrow[r, "f^I"] & I \end{tikzcd}$$

   where ${t_i}^I : I^n \to I$ is the interpretation of the subterm $t_i$, for $i = 1, \ldots, k$, and $f^I$ is the interpretation of the basic operation $f$.

It is clear that the interpretation of a term really depends on the context, and when necessary we shall write $t^I = [\,x_1, \ldots, x_n \mid t\,]^I$. For example, the term $f\,x_1$ is interpreted as a morphism $f^I : I \to I$ in context $x_1$, and as the morphism $f^I \circ \pi_1 : I^2 \to I$ in the context $x_1, x_2$.

Suppose $u$ and $v$ are terms in context $x_1, \ldots, x_n$. Then we say that the equation in context $x_1, \ldots, x_n \mid u = v$ is *satisfied* by the interpretation $I$ if $u^I$ and $v^I$ are the same morphism in $\mathcal{C}$. In particular, if $u = v$ is an axiom of the theory, and $x_1, \ldots, x_n$ are all the variables appearing in either $u$ or $v$, we say that $I$ *satisfies the axiom* $u = v$, written

$$I \models u = v,$$

if $[x_1, \ldots, x_n \mid u]^I$ and $[x_1, \ldots, x_n \mid v]^I$ are the same morphism,

$$\begin{tikzcd} I^n \arrow[r, shift left, "{[x_1, \ldots, x_n \mid u]^I}"] \arrow[r, shift right, "{[x_1, \ldots, x_n \mid v]^I}"'] & I\,. \end{tikzcd} \tag{1.4}$$

We can then define, as expected:

**Definition 1.1.11** (cf. Definition **??**)**.** A *model* $M$ of an algebraic theory $\mathbb{T}$ in a category $\mathcal{C}$ with finite products (also called a $\mathbb{T}$-*algebra*) is an interpretation of the signature $\Sigma_{\mathbb{T}}$,

$$f^I : I^k \longrightarrow I \qquad \text{in } \mathcal{C},$$

for all $f \in \Sigma_{\mathbb{T}}$, that satisfies the axioms $A_{\mathbb{T}}$,

$$I \models u = v,$$

for all $(u = v) \in A_{\mathbb{T}}$.

A *homomorphism* of models $h : M \to N$ is an arrow in $\mathcal{C}$ that commutes with the interpretations of the basic operations,

$$h \circ f^M = f^N \circ h^k$$

for all $f \in \Sigma_{\mathbb{T}}$, as indicated in:

$$\begin{tikzcd}
M^k \arrow[r, "h^k"] \arrow[d, "f^M"'] & N^k \arrow[d, "f^N"] \\
M \arrow[r, "h"'] & N
\end{tikzcd}$$

The category of $\mathbb{T}$-models in $\mathcal{C}$ is written,

$$\mathsf{Mod}(\mathbb{T}, \mathcal{C}).$$

A model of the trivial theory $\mathbb{T}_0$ in $\mathcal{C}$ is therefore just an object $A$ in $\mathcal{C}$, and a homomorphism is just a map, so

$$\mathsf{Mod}(\mathbb{T}_0, \mathcal{C}) = \mathcal{C}.$$

A model of the theory $\mathbb{T}_{\mathsf{Group}}$ of groups in $\mathcal{C}$ is a group in $\mathcal{C}$, in the above sense, and similarly for homomorphisms, so:

$$\mathsf{Mod}(\mathbb{T}_{\mathsf{Group}}, \mathcal{C}) = \mathsf{Group}(\mathcal{C})$$

as defined above. In particular, a model in $\mathsf{Set}$ is just a group in the usual sense, so we have:

$$\mathsf{Mod}(\mathbb{T}_{\mathsf{Group}}, \mathsf{Set}) = \mathsf{Group}(\mathsf{Set}) = \mathsf{Group}.$$

An example of a new kind is provided by the following.

**Example 1.1.12.** A model of the theory of groups in a functor category $\mathsf{Set}^{\mathbb{C}}$ is the same thing as a functor from $\mathbb{C}$ into the category groups,

$$\mathsf{Group}(\mathsf{Set}^{\mathbb{C}}) = \mathsf{Group}(\mathsf{Set})^{\mathbb{C}} \cong \mathsf{Group}^{\mathbb{C}}.$$

Indeed, for each object $C \in \mathbb{C}$ there is an evaluation functor,

$$\mathsf{eval}_C : \mathsf{Set}^{\mathbb{C}} \to \mathsf{Set}$$

with $\mathsf{eval}_C(F) = F(C)$, and evaluation preserves products since these are computed pointwise in the functor category. Moreover, every arrow $h : C \to D$ in $\mathbb{C}$ gives rise to an obvious natural transformation $h : \mathsf{eval}_C \to \mathsf{eval}_D$. Thus for any group $G$ in $\mathsf{Set}^{\mathbb{C}}$, we have groups $\mathsf{eval}_C(G)$ for each $C \in \mathbb{C}$ and group homomorphisms $h_G : C(G) \to D(G)$, comprising a functor $G : \mathbb{C} \to \mathsf{Group}$. Conversely, it is clear that a functor $H : \mathbb{C} \to \mathsf{Group}$ determines a group $H$ in $\mathsf{Set}^{\mathbb{C}}$ with underlying object $|HC|$, where $| - | : \mathsf{Group} \to \mathsf{Set}$ is the forgetful functor. These constructions are clearly mutually inverse (up to canonical isomorphisms determined by the choice of products). Thus, briefly, *a group in the category of variable sets may be regarded as a variable group.*

**Exercise 1.1.13.** Verify the details of the isomorphism of categories

$$\mathsf{Mod}(\mathbb{T}, \mathsf{Set}^{\mathbb{C}}) \cong \mathsf{Mod}(\mathbb{T}, \mathsf{Set})^{\mathbb{C}},$$

as example 1.1.12, for an arbitrary algebraic theory $\mathbb{T}$.

**Exercise 1.1.14.** Determine what a group is in the following categories: the category of graphs $\mathsf{Graph}$, the category of topological spaces $\mathsf{Top}$, and the category of groups $\mathsf{Group}$. (Hint: Only the last case is tricky. Before thinking too hard about it, prove the following lemma [Bor94, Lemma 3.11.6], known as the Eckmann-Hilton argument. Let $G$ be a set provided with two binary operations $\cdot$ and $\star$ and a common unit $e$, so that $x \cdot e = e \cdot x = x \star e = e \star x = x$. Suppose the two operations commute, i.e., $(x \star y) \cdot (z \star w) = (x \cdot z) \star (y \cdot w)$. Then they coincide, and are *commutative* and associative.)

## 1.1.2 Theories as categories

The syntactically presented notion of an algebraic theory is a practical convenience, but as a specification of a mathematical concept, say that of a group, it has some defects. We would prefer a *presentation-free* notion that captures the group concept without tying it to a specific syntactic presentation (the example below indicates why). One such notion can be given by a category with a certain universal property, which determines it uniquely, up to equivalence of categories.

Let us consider group theory again. The algebraic axiomatization in terms of unit, multiplication and inverse is not the only possible one. For example, an alternative formulation uses the unit $e$ and a binary operation $\odot$, called *double division*, along with a single axiom [McC93]:

$$(x \odot (((x \odot y) \odot z) \odot (y \odot e))) \odot (e \odot e) = z \ .$$

The usual group operations are related to double division as follows:

$$x \odot y = x^{-1} \cdot y^{-1} \ , \qquad x^{-1} = x \odot e \ , \qquad x \cdot y = (x \odot e) \odot (y \odot e) \ .$$

There may be practical reasons for prefering one formulation of group theory over another, but this should not determine what the general concept of a group is. For example, we would like to avoid particular choices of basic constants, operations, and axioms. This is akin to the situation where an algebra is presented by generators and relations: the algebra itself is regarded as independent of any particular such presentation. Similarly, one usually prefers a basis-free theory of vector spaces: it is better to formulate the general idea of a vector space without refering explicitly to a basis, even though every vector space has one.

As a first step, one could simply take *all* operations built from unit, multiplication, and inverse as basic, and *all* valid equations of group theory as axioms. But we can go a step further and collect all the operations into a category, thus forgetting about which ones were “basic”, and which equalities were “axioms”. We first describe this construction of a category $\mathcal{C}_{\mathbb{T}}$ for an algebraic theory $\mathbb{T}$, and then determine a universal characterization of it.

As objects of $\mathcal{C}_{\mathbb{T}}$ we take the *contexts*, i.e. sequences of distinct variables,

$$[x_1, \ldots, x_n] \,. \qquad (n \geq 0)$$

Actually, it will be more convenient to take equivalence classes under renaming of variables, so that $[x_1, x_3] = [x_2, x_1]$. That is to say, the objects are just natural numbers; but it will be useful to continue to write them as contexts.

A morphism from $[x_1, \ldots, x_m]$ to $[x_1, \ldots, x_n]$ is then an $n$-tuple $(t_1, \ldots, t_n)$, where each $t_k$ is a term in the context $x_1, \ldots, x_m$, possibly after renaming the variables. Two such morphisms $(t_1, \ldots, t_n)$ and $(s_1, \ldots, s_n)$ are equal if, and only if, the axioms of the theory formally imply that $t_k = s_k$ for every $k = 1, \ldots, n$,

$$\mathbb{T} \vdash t_k = s_k \,.$$

Strictly speaking, morphisms are thus (tuples of) *equivalence classes* of terms in context, written,

$$[x_1, \ldots, x_m \mid t_1, \ldots, t_n\,] : [x_1, \ldots, x_m] \longrightarrow [x_1, \ldots, x_n],$$

where two terms are equivalent when the theory proves them to be equal (after renaming the variables). Since it is rather cumbersome to work with such equivalence classes, we shall work with the terms directly, but keeping in mind that equality between them is this equivalence. Note also that the context of the morphism agrees with its domain, so we can omit it from the notation when that domain is clear. The composition of morphisms

$$\begin{aligned}(t_1, \ldots, t_m) &: [x_1, \ldots, x_k] \to [x_1, \ldots, x_m] \\ (s_1, \ldots, s_n) &: [x_1, \ldots, x_m] \to [x_1, \ldots, x_n]\end{aligned}$$

is the morphism $(r_1, \ldots, r_n)$ whose $i$-th component is obtained by simultaneously substituting in $s_i$ the terms $t_1, \ldots, t_m$ for the variables $x_1, \ldots, x_m$:

$$r_i = s_i[t_1, \ldots, t_m / x_1, \ldots, x_m] \qquad (1 \leq i \leq n)$$

The identity morphism on the object $[x_1, \ldots, x_n]$ is the equivalence class of $(x_1, \ldots, x_n)$.

Using the usual rules of deduction for equational logic (see Section A.5), it is easy to verify that these specifications are well-defined on equivalence classes, and therefore make $\mathcal{C}_{\mathbb{T}}$ a category.

**Definition 1.1.15.** The category $\mathcal{C}_{\mathbb{T}}$ just defined is called the *syntactic category* of the algebraic theory $\mathbb{T}$.

The syntactic category $\mathcal{C}_{\mathbb{T}}$ (which may be thought of as the "Lindenbaum-Tarski category" of $\mathbb{T}$, see **??**) contains the same "algebraic" information as the theory $\mathbb{T}$ from which it was built, but in a syntax-invariant way. Two different syntactic presentations of $\mathbb{T}$ — like the ones for groups mentioned above — will give rise to essentially the same category $\mathcal{C}_{\mathbb{T}}$ (i.e. up to isomorphism). In this sense, the category $\mathcal{C}_{\mathbb{T}}$ is the abstract, algebraic object presented by the "generators and relations" (the operations and equations) of $\mathbb{T}$. But there is another, still more important, sense in which $\mathcal{C}_{\mathbb{T}}$ represents $\mathbb{T}$, as we next show.

**Exercise 1.1.16.** Show that the syntactic category $\mathcal{C}_{\mathbb{T}}$ has all finite products.

### 1.1.3 Models as functors

Having represented an algebraic theory $\mathbb{T}$ by the syntactic category $\mathcal{C}_{\mathbb{T}}$ constructed from it, we next show that $\mathcal{C}_{\mathbb{T}}$ has the universal property that models of $\mathbb{T}$ correspond uniquely to certain functors from $\mathcal{C}_{\mathbb{T}}$. More precisely, given any category with finite products $\mathcal{C}$ (which we shall call an *FP-category*), there is a natural (in $\mathcal{C}$) equivalence,

$$\frac{\mathcal{M} \in \mathsf{Mod}(\mathbb{T}, \mathcal{C})}{M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}} \tag{1.5}$$

between models $\mathcal{M}$ of $\mathbb{T}$ in $\mathcal{C}$ and finite product preserving functors ("*FP-functors*") $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$. The equivalence is mediated by a "universal model" $\mathcal{U}$ in $\mathcal{C}_{\mathbb{T}}$, corresponding to the identity functor $1_{\mathcal{C}_{\mathbb{T}}} : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}_{\mathbb{T}}$ under the above displayed equivalence. By naturality, every model $\mathcal{M}$ then arises as the functorial image $M(\mathcal{U}) \cong \mathcal{M}$ of $\mathcal{U}$ under an essentially unique FP-functor $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$. The possibility of such universal models is an advantage of the generalized notion of a model in a category other than $\mathsf{Set}$.

To give the details of the correspondence (1.5), let $\mathbb{T}$ be an arbitrary algebraic theory and $\mathcal{C}_{\mathbb{T}}$ the syntactic category constructed from $\mathbb{T}$ as in Definition 1.1.15. It is easy to show that the product in $\mathcal{C}_{\mathbb{T}}$ of two objects $[x_1, \dots, x_n]$ and $[x_1, \dots, x_m]$ is the object $[x_1, \dots, x_{n+m}]$, and that $\mathcal{C}_{\mathbb{T}}$ has all finite products, including $\mathsf{1} = [-]$, the empty context (see Exercise 1.1.16). Moreover, there is a distinguished $\mathbb{T}$-model $\mathcal{U}$ in $\mathcal{C}_{\mathbb{T}}$ consisting of the signature $\Sigma_{\mathbb{T}}$ itself, which we call the *syntactic model*: the underlying object $U = |\mathcal{U}|$ is the context $[x_1]$ of length one, and each operation symbol $f$, of say arity $k$, is interpreted as itself,

$$f^{\mathcal{U}} = [\, f(x_1, \dots, x_k)\,] : U^k = [x_1, \dots, x_k] \longrightarrow [x_1] = U.$$

The axioms are all satisfied, because the equivalence relation on terms is determined by $\mathbb{T}$-provability. Explicitly, for all terms $s, t$ we have:

$$\mathcal{U} \models s = t \quad \Longleftrightarrow \quad s^{\mathcal{U}} = t^{\mathcal{U}} \quad \Longleftrightarrow \quad \mathbb{T} \vdash s = t. \tag{1.6}$$

We record this fact as the following.

**Proposition 1.1.17.** *The syntactic model $\mathcal{U}$ in $\mathcal{C}_{\mathbb{T}}$ is "logically generic" in the sense that it satisfies all and only the $\mathbb{T}$-provable equations, as in* (1.1.17)*.*

*Proof.* A rigorous proof requires that one show that every term $t$ is interpreted in $\mathcal{U}$ by "itself", i.e. by its own equivalence class under $\mathbb{T}$-provable equality,

$$[\, x_1, \dots, x_m \mid t\,]^{\mathcal{U}} = [\, x_1, \dots, x_m \mid t\,]$$

□

Even more important than being logically generic, though, is the following universal property of the syntactic model $\mathcal{U}$ in $\mathcal{C}_{\mathbb{T}}$.

> The *universal property* of $\mathcal{C}_{\mathbb{T}}$ with the syntactic model $\mathcal{U}$ is as follows: Any model $\mathcal{M}$ in any finite product category $\mathcal{C}$ is the image of $\mathcal{U}$ under an essentially unique, finite product preserving functor $\mathcal{M}^{\sharp} : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$.

(See Definition 1.1.20 below for a more precise formulation.) In this sense, the syntactic category $\mathcal{C}_{\mathbb{T}}$ may be thought of as the free finite product category with a model of $\mathbb{T}$. To show this formally, first observe that any FP-functor $F : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ takes the syntactic model $\mathcal{U}$ in $\mathcal{C}_{\mathbb{T}}$ to a model $F\mathcal{U}$ in $\mathcal{C}$, with interpretations

$$f^{F\mathcal{U}} = Ff^{\mathcal{U}} : FU^k \to FU \qquad \text{for each } f \in \Sigma_k.$$

Indeed, that is true for any FP-category $\mathcal{D}$ in place of $\mathcal{C}_{\mathbb{T}}$ and any model in $\mathcal{D}$. Similarly, any natural transformation $\vartheta : F \to G$ between FP-functors determines a homomorphism of models $h = \vartheta_{\mathcal{U}} : F\mathcal{U} \to G\mathcal{U}$. In more detail, suppose $f : U \times U \to U$ is a basic operation, then there is a commutative diagram,

$$\begin{tikzcd}
FU \times FU \arrow[r, "h \times h"] \arrow[d, "\cong"] \arrow[dd, bend right=60, "f^{F\mathcal{U}}"'] & GU \times GU \arrow[d, "\cong"'] \arrow[dd, bend left=60, "f^{G\mathcal{U}}"] \\
F(U \times U) \arrow[r, "\vartheta_{U \times U}"] \arrow[d, "Ff"] & G(U \times U) \arrow[d, "Gf"'] \\
FU \arrow[r, "h = \vartheta_U"'] & GU
\end{tikzcd}$$

where the upper square commutes by preservation of products, and the lower one by naturality. Thus the operation "evaluation at $\mathcal{U}$" always determines a functor,

$$\mathsf{eval}_{\mathcal{U}} : \mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{1.7}$$

from the category of finite product preserving functors $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}$, with natural transformations as arrows, into the category of $\mathbb{T}$-models in $\mathcal{C}$. This much is also true for any model in any FP-category $\mathcal{D}$; what is special about $\mathcal{U}$ is the following.

**Proposition 1.1.18.** *The functor* (1.7) *is an equivalence of categories, natural in* $\mathcal{C}$.

*Proof.* Let $\mathcal{M}$ be any model in an FP-category $\mathcal{C}$. Then the assignment $f \mapsto f^{\mathcal{M}}$ given by the interpretation part of $\mathcal{M}$ determines a functor $\mathcal{M}^{\sharp} : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$, defined on objects by

$$\mathcal{M}^{\sharp}[x_1, \dots, x_k] = M^k$$

and on morphisms by

$$\mathcal{M}^\sharp(t_1, \ldots, t_n) = \big({t_1}^{\mathcal{M}}, \ldots, {t_n}^{\mathcal{M}}\big)\,.$$

In more detail, $\mathcal{M}^\sharp$ is defined on morphisms

$$[x_1, \ldots, x_k \mid t\,] : [x_1, \ldots, x_k] \to [x_1, \ldots, x_n]$$

in $\mathcal{C}_\mathbb{T}$ by the following rules:

1. The morphism

   $$(x_i) : [x_1, \ldots, x_k] \to [x_1]$$

   is mapped to the $i$-th projection

   $$\pi_i : M^k \to M.$$

2. The morphism

   $$f(t_1, \ldots, t_m) : [x_1, \ldots, x_k] \to [x_1]$$

   is mapped to the composite

   $$\begin{tikzcd} M^k \arrow[r, "\big(\mathcal{M}^\sharp t_1, \ldots, \mathcal{M}^\sharp t_m\big)"] & M^m \arrow[r, "f^{\mathcal{M}}"] & M \end{tikzcd}$$

   where $\mathcal{M}^\sharp t_i : M^k \to M$ is the value of $\mathcal{M}^\sharp$ on the morphisms $(t_i) : [x_1, \ldots, x_k] \to [x_1]$, for $i = 1, \ldots, m$, and $f^{\mathcal{M}}$ is the interpretation of the basic operation $f$.

3. The morphism

   $$(t_1, \ldots, t_n) : [x_1, \ldots, x_k] \to [x_1, \ldots, x_n]$$

   is mapped to the morphism $\big(\mathcal{M}^\sharp t_1, \ldots, \mathcal{M}^\sharp t_n\big)$ where $\mathcal{M}^\sharp t_i$ is the value of $\mathcal{M}^\sharp$ on the morphism $(t_i) : [x_1, \ldots, x_k] \to [x_1]$, and

   $$\big(\mathcal{M}^\sharp t_1, \ldots, \mathcal{M}^\sharp t_n\big) : M^k \longrightarrow M^n$$

   is the evident $n$-tuple in the FP-category $\mathcal{C}$.

That $\mathcal{M}^\sharp : \mathcal{C}_\mathbb{T} \to \mathcal{C}$ really is a functor now follows from the assumption that the interpretation $M$ is a model, meaning that all the equations of the theory are satisfied by it, so that the above specification is well-defined on equivalence classes. Note that the functor $\mathcal{M}^\sharp$ is defined in such a way that it obviously preserves finite products, and that, moreover, there is an isomorphism of models,

$$\mathcal{M}^\sharp(\mathcal{U}) \cong \mathcal{M}.$$

Thus we have shown that the functor "evaluation at $\mathcal{U}$",

$$\mathsf{eval}_\mathcal{U} : \mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_\mathbb{T}, \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{1.8}$$

is essentially surjective on objects, since $\mathsf{eval}_\mathcal{U}(\mathcal{M}^\sharp) = \mathcal{M}^\sharp(\mathcal{U}) \cong \mathcal{M}$.

We leave the verification that it is full and faithful as an easy exercise.

**Exercise 1.1.19.** Verify this.

Finally, naturality in $\mathcal{C}$ means the following. Suppose $\mathcal{M}$ is a model of $\mathbb{T}$ in any FP-category $\mathcal{C}$. Any FP-functor $F : \mathcal{C} \to \mathcal{D}$ to another FP-category $\mathcal{D}$ then takes $\mathcal{M}$ to a model $F(\mathcal{M})$ in $\mathcal{D}$. Just as for the special case of $\mathcal{U}$, the interpretation is given by setting $f^{F(\mathcal{M})} = F(f^{\mathcal{M}})$ for the basic operations $f$ (and composing with the canonical isos coming from preservation of products, $F(M) \times F(M) \cong F(M \times M)$, etc.). Since equations are described by commuting diagrams, $F$ takes a model to a model, and the same is true for homomorphisms. Thus $F : \mathcal{C} \to \mathcal{D}$ induces a functor on $\mathbb{T}$-models,

$$\mathsf{Mod}(\mathbb{T}, F) : \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{D}).$$

By naturality of (1.7), we mean that the following square commutes up to natural isomorphism:

$$\begin{tikzcd}
\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \arrow[r, "\mathsf{eval}_U"] \arrow[d, swap, "\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, F)"] & \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \arrow[d, "\mathsf{Mod}(\mathbb{T}, F)"] \\
\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{D}) \arrow[r, swap, "\mathsf{eval}_U"] & \mathsf{Mod}(\mathbb{T}, \mathcal{C})
\end{tikzcd} \tag{1.9}$$

But this is clear, since for any FP-functor $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ we have:

$$\begin{aligned}
\mathsf{eval}_{\mathcal{U}} \circ \mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, F)(M) &= (\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, F)(M))(\mathcal{U}) \\
&= (F \circ M)(\mathcal{U}) \\
&= F(M(\mathcal{U})) \\
&= F(\mathsf{eval}_{\mathcal{U}}(M)) \\
&\cong \mathsf{Mod}(\mathbb{T}, F)(\mathsf{eval}_{\mathcal{U}}(M)) \\
&= \mathsf{Mod}(\mathbb{T}, F) \circ \mathsf{eval}_{\mathcal{U}}(M).
\end{aligned}$$

□

The equivalence of categories

$$\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \simeq \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{1.10}$$

actually determines $\mathcal{C}_{\mathbb{T}}$ and the universal model $\mathcal{U}$ uniquely, up to equivalence of categories and isomorphism of models. Indeed, to recover $\mathcal{U}$, just put $\mathcal{C}_{\mathbb{T}}$ for $\mathcal{C}$ and the identity functor $1_{\mathcal{C}_{\mathbb{T}}}$ on the left, to get $\mathcal{U}$ in $\mathsf{Mod}(\mathbb{T}, \mathcal{C}_{\mathbb{T}})$ on the right! To see that $\mathcal{C}_{\mathbb{T}}$ itself is also determined, observe that (1.10) says that the functor $\mathsf{Mod}(\mathbb{T}, \mathcal{C})$ is *representable*, with representing object $\mathcal{C}_{\mathbb{T}}$, in an appropriate (i.e. bicategorical) sense. As usual, this fact can also be formulated in elementary terms as a universal mapping property of $\mathcal{C}_{\mathbb{T}}$, as follows:

**Definition 1.1.20.** The *classifying category* of an algebraic theory $\mathbb{T}$ is an FP-category $\mathcal{C}_{\mathbb{T}}$ with a distinguished model $\mathcal{U}$, called the *universal model*, such that:

(i) for any model $\mathcal{M}$ in any FP-category $\mathcal{C}$, there is an FP-functor

$$\mathcal{M}^{\sharp} : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$$

and an isomorphism of models $\mathcal{M} \cong \mathcal{M}^{\sharp}(\mathcal{U})$.

(ii) for any FP-functors $F, G : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ and model homomorphism $h : F(\mathcal{U}) \to G(\mathcal{U})$, there is a unique natural transformation $\vartheta : F \to G$ with

$$\vartheta_{\mathcal{U}} = h.$$

Observe that (i) says that the evaluation functor (1.7) is essentially surjective, and (ii) that it is full and faithful. The category $\mathcal{C}_{\mathbb{T}}$ is then determined, up to equivalence, by this universal mapping property. Specifically, if $(\mathcal{C}, \mathcal{U})$ and $(\mathcal{D}, \mathcal{V})$ are both classifying categories for the same theory, then there are classifying functors,

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "\mathcal{V}^{\sharp}"] & \mathcal{D} \arrow[l, bend left, "\mathcal{U}^{\sharp}"]
\end{tikzcd}$$

the composites of which are necessarily isomorphic to the respective identity functors, since e.g. $\mathcal{U}^{\sharp}(\mathcal{V}^{\sharp}(\mathcal{U})) \cong \mathcal{U}^{\sharp}(\mathcal{V}) \cong \mathcal{U}$.

We have now shown not only that every algebraic theory has a classifying category, but also that the syntactic category is essentially determined by that distinguishing property. We record this as the following.

**Theorem 1.1.21.** *Every algebraic theory $\mathbb{T}$ has a classifying category $\mathcal{C}_{\mathbb{T}}$, which can be constructed as the syntactic category of $\mathbb{T}$, in the sense of Definition 1.1.15.*

**Example 1.1.22.** Let us see explicitly what the foregoing definitions give us in the case of the theory of groups $\mathbb{T}_{\mathsf{Group}}$. Let us write $\mathbb{G} = \mathcal{C}_{\mathbb{T}_{\mathsf{Group}}}$ for the classifying category, which has contexts $[x_1, \dots, x_n]$ as objects, and terms built from variables and the group operations (modulo renaming of variables and provability from the group laws) as arrows. A finite product preserving functor $G : \mathbb{G} \to \mathsf{Set}$ is determined uniquely, up to natural isomorphism, by its action on the context $[x_1]$ and the terms representing the basic operations. If we set

$$\begin{aligned}
|\mathcal{G}| &:= G[x_1] \,, & u^{\mathcal{G}} &:= G(\cdot \mid e) \,, \\
i^{\mathcal{G}} &:= G(x_1 \mid {x_1}^{-1}) \,, & m_G &= G(x_1, x_2 \mid x_1 \cdot x_2) \,,
\end{aligned}$$

then $\mathcal{G} = (|\mathcal{G}|, u^{\mathcal{G}}, i^{\mathcal{G}}, m^{\mathcal{G}})$ is just a group, with unit $u^{\mathcal{G}}$, inverse $i^{\mathcal{G}}$, and multiplication $m^{\mathcal{G}}$. That $\mathcal{G}$ satisfies the axioms for groups follows from the functoriality of $G$ and preservation of finite products, which implies preservation of the corresponding commutative diagrams. Conversely, any group $\mathcal{G} = (G, u, i, m)$ determines a finite product preserving functor $\mathcal{G}^{\sharp} : \mathbb{G} \to \mathsf{Set}$, by setting $\mathcal{G}^{\sharp}[x_1] = G$, etc. Thus $\mathsf{Mod}(\mathbb{G}, \mathsf{Set})$ will indeed be equivalent to $\mathsf{Group}$ once we show that both categories have the same notion of morphisms. This is shown just as in the general case above.

**Example 1.1.23.** Recall from 1.1.12 that a group $G$ in the functor category $\mathsf{Set}^{\mathbb{C}}$ is essentially the same thing as a functor $G : \mathbb{C} \to \mathsf{Group}$. From the point of view of algebras as functors, this amounts to the observation that product-preserving functors $\mathbb{G} \to \mathsf{Hom}(\mathbb{C}, \mathsf{Set})$ correspond (by exponential transposition) to functors $\mathbb{C} \to \mathsf{Hom}_{\mathsf{FP}}(\mathbb{G}, \mathsf{Set})$, where the latter $\mathsf{Hom}$-set consists just of product-preserving functors. The correspondence extends to natural transformations, giving the previously observed (Example 1.1.12) equivalance of categories,

$$\mathsf{Group}(\mathsf{Set}^{\mathbb{C}}) \simeq \mathsf{Group}(\mathsf{Set})^{\mathbb{C}} = \mathsf{Group}^{\mathbb{C}}.$$

### 1.1.4 Soundness and completeness

Consider an algebraic theory $\mathbb{T}$ and an equation $s = t$ between terms of the theory. If the equation can be proved from the axioms of the theory, $\mathbb{T} \vdash s = t$, then every model $\mathcal{M}$ of the theory in any FP-category satisfies the equation, $\mathcal{M} \models s = t$. This is called the *soundness* of the equational calculus with respect to categorical models, and it can be shown by a straightforward induction on the equational proof that establishes $\mathbb{T} \vdash s = t$. The converse statement reads:

$$\mathcal{M} \models s = t, \text{ for all } \mathcal{M} \quad \Rightarrow \quad \mathbb{T} \vdash s = t \,.$$

This is called *completeness*, and (together with soundness) it says that the equational calculus suffices for proving all (and only) the equations that hold generally in the semantics. For functorial semantics, this condition holds in an especially strong way: by Proposition 1.1.17, we already know that the syntactic model $\mathcal{U}$ in $\mathcal{C}_{\mathbb{T}}$ is logically generic, in the sense that satisfaction by $\mathcal{U}$ is equivalent to provability in $\mathbb{T}$,

$$\mathcal{U} \models s = t \iff \mathbb{T} \vdash s = t.$$

But since $\mathcal{U}$ is also universal in the sense of Definition 1.1.20, it follows immediately that we also have soundness and completeness:

**Theorem 1.1.24** (Soundness and completeness of equational logic)**.** *For any terms $s, t$ we have $\mathbb{T} \vdash s = t$ if and only if every model $\mathcal{M}$ in every FP-category $\mathcal{C}$ satisfies $s = t$.*

*Proof.* If $\mathbb{T} \vdash s = t$, then by Proposition 1.1.17 (the syntactic construction of $\mathcal{C}_{\mathbb{T}}$) we have $\mathcal{U} \models s = t$, meaning that $s^{\mathcal{U}} = t^{\mathcal{U}}$. But then for any model $\mathcal{M}$ in an FP-category $\mathcal{C}$, we obtain $\mathcal{M} \models s = t$ by applying the classifying functor $\mathcal{M}^\sharp : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$, which preserves the interpretations of $s$ and $t$,

$$\mathcal{M}^\sharp(s^{\mathcal{U}}) = s^{\mathcal{M}^\sharp(\mathcal{U})} = s^{\mathcal{M}}$$

and so from $s^{\mathcal{U}} = t^{\mathcal{U}}$ we get $s^{\mathcal{M}} = t^{\mathcal{M}}$.

Conversely, if $\mathcal{M} \models s = t$ for every model $\mathcal{M}$, then in particular $\mathcal{U} \models s = t$, and so $\mathbb{T} \vdash s = t$, since $\mathcal{U}$ is generic. □

Classically, it is seldom the case that there exists a generic model; instead, one usually considers the class of all models in, say, $\mathsf{Set}$. Completeness with respect to a restricted class of models is of course a stronger statement than completeness with respect to all models in all categories. Toward the classical result, we can first consider completeness with respect to "variable models" in $\mathsf{Set}$, i.e. in arbitrary presheaf categories $\mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$.

**Lemma 1.1.25.** *Let $\mathbb{T}$ be an algebraic theory. The Yoneda embedding*

$$\mathsf{y} : \mathcal{C}_{\mathbb{T}} \to \widehat{\mathcal{C}_{\mathbb{T}}}$$

*is a generic model for $\mathbb{T}$.*

*Proof.* The Yoneda embedding $\mathsf{y} : \mathcal{C}_{\mathbb{T}} \to \widehat{\mathcal{C}_{\mathbb{T}}}$ preserves limits, and in particular finite products, hence it determines a model $\mathcal{Y} = \mathsf{y}\mathcal{U}$ in the category of presheaves $\widehat{\mathcal{C}_{\mathbb{T}}}$. Like all models, $\mathcal{Y}$ satisfies all the equations that hold in $\mathcal{U}$, simply because $\mathsf{y}$ is an FP functor. But because $\mathsf{y}$ is also faithful, any equation that holds in $\mathcal{Y}$ must already hold in $\mathcal{U}$, and therewith in all models. □

**Example 1.1.26.** We consider group theory one last time. As a presheaf on (the classifying category of) the theory of groups $\mathbb{G}$, the generic group $\mathcal{Y}$ satisfies every equation that is satisfied by all groups, and no others. Let us describe its underlying object $Y = |\mathcal{Y}|$ explicitly as a "variable set". The presheaf $Y$ is represented by the context with one variable,

$$Y = \mathsf{y}[x_1] = \mathbb{G}(-, [x_1]) \; .$$

The values of this functor thus comprise a family of sets parametrized by the objects $[x_1, \dots, x_n]$ of $\mathbb{G}$; namely, for every $n \in \mathbb{N}$, we have the set

$$Y_n = \mathbb{G}([x_1, \dots, x_n], [x_1])$$

which consists of all terms in $n$ variables, modulo the equations of group theory; but this is just the set of elements of the *free group* $F(n)$ on $n$ generators! Thus we have

$$Y_n = \mathbb{G}([x_1, \dots, x_n], [x_1]) \cong |F(n)| \cong \mathsf{Set}(1, |F(n)|) \cong \mathsf{Group}(F(1), F(n)).$$

Moreover, the unit, inverse, and multiplication operations on $Y$ agree at each stage $Y_n$ with the operations on the free group $F(n)$ (as the reader should verify!).

To summarize, the presheaf of groups $\mathcal{Y} : \mathbb{G}^{\mathsf{op}} \to \mathsf{Group}$ on the theory $\mathbb{G}$ of groups is isomorphic to the functor $F : \mathbb{G}^{\mathsf{op}} \to \mathsf{Group}$ of free groups $F(n)$ on $n$-generators (at least pointwise). We will see why this is so in more detail in section **??** .

Finally, we consider the completeness of equational logic with respect to all $\mathsf{Set}$-valued models $\mathcal{M} : \mathcal{C}_{\mathbb{T}} \to \mathsf{Set}$, which of course correspond to classical $\mathbb{T}$-algebras. We need the following:

**Lemma 1.1.27.** *For any small category $\mathbb{C}$, there is a jointly faithful family $(E_i)_{i \in I}$ of FP-functors $E_i : \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}} \to \mathsf{Set}$, with $I$ a set. That is, for any maps $f, g : A \to B$ in $\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$, if $E_i(f) = E_i(g)$ for all $i \in I$, then $f = g$.*

*Proof.* We can take $I = \mathbb{C}_0$ and the evaluation functors $E_C = \mathsf{eval}_C : \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}} \to \mathsf{Set}$, for all $C \in \mathbb{C}$. These are clearly jointly faithful. Note that they also preserve all limits and colimits, which are constructed pointwise in presheaves. □

**Proposition 1.1.28.** *Suppose $\mathbb{T}$ is an algebraic theory. For any terms $s, t$,*

$$\mathcal{M} \models s = t \quad \textit{for all models } \mathcal{M} \textit{ in } \mathsf{Set} \iff \mathbb{T} \vdash s = t.$$

*Thus the equational logic of algebraic theories is sound and complete with respect to* Set*-valued semantics.*

*Proof.* Combine the foregoing lemma with the fact, from Lemma 1.1.25, that the Yondea embedding is a generic model. □

The completeness of equational reasoning was originally proved by Birkhoff [**?**]. The proof is not particularly difficult; we have chosen to redo it in this way because the method will generalize to other systems of logic in later chapters.

**Exercise 1.1.29.** We described the object part of the functor $Y = \mathsf{y}U : \mathbb{G}^{\mathsf{op}} \to \mathsf{Set}$ represented by the underlying object $U = [x_1]$ of the universal group $\mathcal{U}$, in terms of the free groups $F(n)$. What is the action of $Y$ on the arrows of $\mathbb{G}$ in these terms? Also describe the group structure on $Y$ in $\widehat{\mathbb{G}}$ explicitly.

**Exercise 1.1.30.** Let $t = t(x_1, \ldots, x_n)$ be a term of group theory in the variables $x_1, \ldots, x_n$. On the one hand we can think of $t$ as an element of the free group $F(n)$, and on the other we can consider the interpretation of $t$ with respect to the representable group $\mathcal{Y}$ in $\widehat{\mathbb{G}}$, namely as a natural transformation $t^{\mathcal{Y}} : Y^n \Longrightarrow Y$. Suppose $s = s(x_1, \ldots, x_n)$ is another such term in the same variables $x_1, \ldots, x_n$. Show that $s^{\mathcal{Y}} = t^{\mathcal{Y}}$ if, and only if, $s = t$ in the free group $F(n)$.

### 1.1.5 Functorial semantics

Let us summarize our treatment of algebraic theories thus far. We have reformulated certain traditional *logical* notions in terms of *categorical* ones. The traditional approach may be described as involving the following four different parts:

*Terms*

There is an underlying *type theory* consisting of types and terms. For algebraic theories there is only one type, which is not even explicitly mentioned. The terms are built from variables and some basic operation symbols.

*Equations*

Algebraic theories have a particularly simple *logic* that involves only equations between terms and equational reasoning, which is basically substitution of equals for equals.

*Theories*

An *algebraic theory* is given by a set of basic terms, the signature, and a set of axioms, which are just equations between terms.

*Models*

Algebraic theories are interpreted as sets equipped with operations. An interpretation is a *model* if it satisfies all the axioms of the theory, which just means that the functions interpreting the terms that occur in the axioms are actually equal.

The alternative approach of *functorial semantics* developed here may be summarized as follows:

*Theories are categories*

From a given theory we construct a structured category, which captures the same information in a way that does not depend on a particular presentation by basic operations and axioms.

*Models are functors*

A model is a structure-preserving functor from the theory to a category with the same structure. For algebraic theories, a model is required to preserves finite products, which along with functoriality ensures that all valid equations of the theory are preserved, and the axioms are therefore satisfied.

*Homomorphisms are natural transformations*

We obtain the notion of a homomorphism of models for free: since models are functors, homomorphisms between them are natural transformations. Such homomorphisms between models of algebraic theories agree with the usual notion of a homomorphism as a function that "respects" the algebraic structure.

*Universal models*

By admitting models in categories other than $\mathsf{Set}$, functorial semantics admits *universal models*: a model $\mathcal{U}$ in the classifying category $\mathcal{C}_{\mathbb{T}}$, such that any model anywhere is a functorial image of $\mathcal{U}$ by an essentially unique, structure-preserving functor. Such a universal model is then *logically generic*, in the sense that it has all and only those logical properties that are had by all models, since such properties are preserved by the functors in question.

*Logical completeness*

The construction of the classifying category $\mathcal{C}_{\mathbb{T}}$ from the syntax of the theory $\mathbb{T}$ implies the *soundness and completeness* of the logic with respect to general categorical semantics. Completeness with respect to a restricted class of models, such as the $\mathsf{Set}$-valued ones, results from an embedding theorem for the classifying category.

# Appendix A

# Logic

## A.1 Concrete and abstract syntax

By *syntax* we generally mean manipulation of finite strings of symbols according to given *grammatical rules*. For instance, the strings "7)6 + /(8" and "$(6 + 8)/7$" both consist of the same symbols but you will recognize one as junk and the other as *well formed* because you have (implicitly) applied the grammatical rules for arithmetical expressions.

Grammatical rules are usually quite complicated, as they need to prescribe associativity of operators (does "$5+6+7$" mean "$(5+6)+7$" or "$5+(6+7)$"?) and their precedence (does "$6+8/7$" mean "$(6+8)/7$" or "$6+(8/7)$"?), the role of *white space* (empty space between symbols and line breaks), rules for nesting and balancing parentheses, etc. It is not our intention to dwell on such details, but rather to focus on the mathematical nature of well-formed expressions, namely that they represent inductively generated finite trees.[1] Under this view the string "$(6+8)/7$" is just a concrete representation of the tree depicted in Figure A.1.

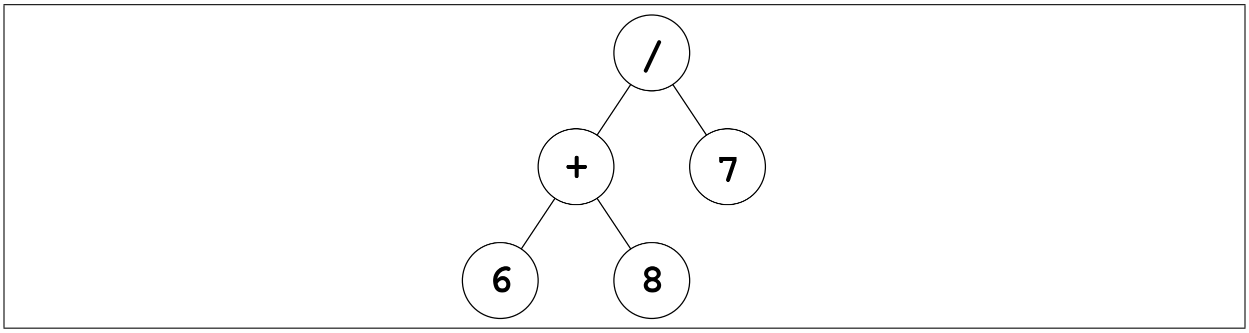

**Figure A.1:** The tree represented by $(6 + 8)/7$

Concrete representation of expressions as finite strings of symbols is called *concrete syntax*, while in *abstract syntax* we view expressions as finite trees. The passage from the

[1] We are limiting attention to the so-called *context-free* grammar, which are sufficient for our purposes. More complicated grammars are rarely used to describe formal languages in logic and computer science.

former to the latter is called *parsing* and is beyond the scope of this book. We will always specify only abstract syntax and assume that the corresponding concrete syntax follows the customary rules for parentheses, associativity and precedence of operators.

As an illustration we give rules for the (abstract) syntax of propositional calculus in *Backus-Naur* form:

$$\text{Propositional variable } p ::= \mathtt{p}_1 \mid \mathtt{p}_2 \mid \mathtt{p}_3 \mid \cdots$$
$$\text{Propositional formula } \phi ::= p \mid \bot \mid \top \mid \phi_1 \land \phi_2 \mid \phi_1 \lor \phi_2 \mid \phi_1 \Rightarrow \phi_2 \mid \neg\phi$$

The vertical bars should be read as "or". The first rule says that a propositional variable is the constant $\mathtt{p}_1$, or the constant $\mathtt{p}_2$, or the constant $\mathtt{p}_3$, etc.[2] The second rule tells us that there are seven inductive rules for building a propositional formula:

- a propositional variable is a formula,
- the constants $\bot$ and $\top$ are formulas,
- if $\phi_1$, $\phi_2$, and $\phi$ are formulas, then so are $\phi_1 \land \phi_2$, $\phi_1 \lor \phi_2$, $\phi_1 \Rightarrow \phi_2$, and $\neg\phi$.

Even though abstract syntax rules say nothing about parentheses or operator associativity and precedence, we shall rely on established conventions for mathematical notation and write down concrete representations of propositional formulas, e.g., $\mathtt{p}_4 \land (\mathtt{p}_1 \lor \neg \mathtt{p}_1) \land \mathtt{p}_4 \lor \mathtt{p}_2$.

A word of warning: operator associativity in syntax is not to be confused with the usual notion of associativity in mathematics. We say that an operator $\star$ is *left associative* when an expression $x \star y \star z$ represents the left-hand tree in Figure A.2, and *right associative* when it represents the right-hand tree. Thus the usual operation of subtraction $-$ is left associative, but is not associative in the usual mathematical sense.

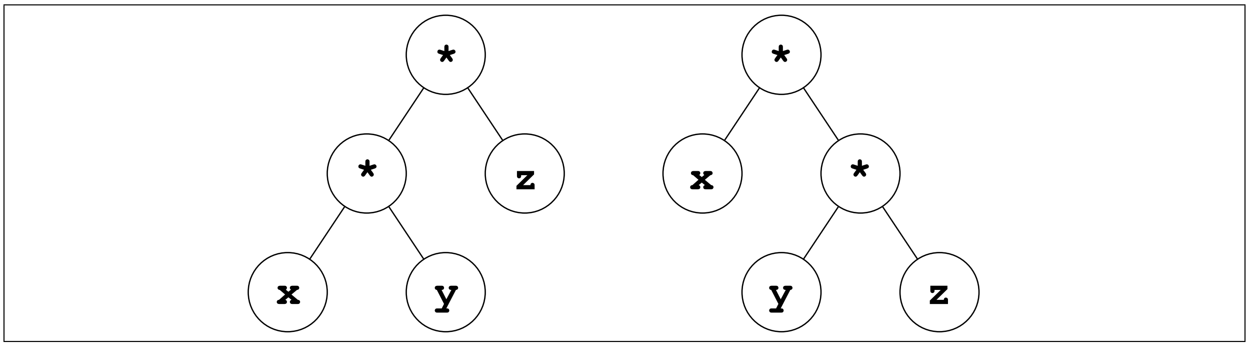

**Figure A.2:** Left and right associativity of $x \star y \star z$

[2] In an actual computer implementation we would allow arbitrary finite strings of letters as propositional variables. In logic we only care about the fact that we can never run out of fresh variables, i.e., that there are countably infinitely many of them.

## A.2 Free and bound variables

Variables appearing in an expression may be *free* or *bound*. For example, in expressions

$$\int_0^1 \sin(a \cdot x)\,dx, \qquad x \mapsto ax^2 + bx + c, \qquad \forall\, x\,.\,(x < a \lor x > b)$$

the variables $a$, $b$ and $c$ are free, while $x$ is bound by the integral operator $\int$, the function formation $\mapsto$, and the universal quantifier $\forall$, respectively. To be quite precise, it is an *occurrence* of a variable that is free or bound. For example, in expression $\phi(x) \lor \exists\, x\,.\,A\psi(x, x)$ the first occurrence of $x$ is free and the remaining ones are bound.

In this book the following operators bind variables:

- quantifiers $\exists$ and $\forall$, cf. **??**,
- $\lambda$-abstraction, cf. **??**,
- search for others **??**.

When a variable is bound we may always rename it, provided the renaming does not confuse it with another variable. In the integral above we could rename $x$ to $y$, but not to $a$ because the binding operation would *capture* the free variable $a$ to produce the unintended $\int_0^1 \sin(a^2)\,da$. Renaming of bound variables is called *$\alpha$-renaming.*

We consider two expressions *equal* if they only differ in the names of bound variables, i.e., if one can be obtained from the other by $\alpha$-renaming. Furthermore, we adhere to *Barendregt's variable convention [***?***, p. 2]*, which says that bound variables are always chosen so as to differ from free variables. Thus we would never write $\phi(x) \lor \exists\, x\,.\,A\psi(x, x)$ but rather $\phi(x) \lor \exists\, y\,.\,A\psi(y, y)$. By doing so we need not worry about capturing or otherwise confusing free and bound variables.

In logic we need to be more careful about variables than is customary in traditional mathematics. Specifically, we always specify which free variables may appear in an expression.[3] We write

$$x_1 : A_1, \ldots, x_n : A_n \mid t$$

to indicate that expression $t$ may contain only free variables $x_1, \ldots, x_n$ of types $A_1, \ldots, A_n$. The list

$$x_1 : A_1, \ldots, x_n : A_n$$

is called a *context* in which $t$ appears. To see why this is important consider the different meaning that the expression $x^2 + y^2 \leq 1$ recevieves in different contexts:

- $x : \mathbb{Z}, y : \mathbb{Z} \mid x^2 + y^2 \leq 1$ denotes the set of tuples $\{(-1, 0), (0, 1), (1, 0), (0, -1)\}$,
- $x : \mathbb{R}, y : \mathbb{R} \mid x^2 + y^2 \leq 1$ denotes the closed unit disc in the plane, and

[3] This is akin to one of the guiding principles of good programming language design, namely, that all variables should be *declared* before they are used.

- $x : \mathbb{R}, y : \mathbb{R}, z : \mathbb{R} \mid x^2 + y^2 \leq 1$ denotes the infinite cylinder in space whose base is the closed unit disc.

In single-sorted theories there is only one type or sort $A$. In this case we abbreviate a context by listing just the variables, $x_1, \ldots, x_n$.

## A.3 Substitution

Substitution is a basic syntactic operation which replaces (free occurrences of) distinct variables $x_1, \ldots, x_n$ in an expression $t$ with expressions $t_1, \ldots, t_n$, which is written as

$$t[t_1/x_1, \ldots, t_n/x_n].$$

We sometimes abbreviate this as $t[\vec{t}/\vec{x}]$ where $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{t} = (t_1, \ldots, t_n)$. Here are several examples:

$$\begin{aligned}(x^2 + x + y)[(2+3)/x] &= (2+3)^2 + (2+3) + y \\ (x^2 + y)[y/x, x/y] &= y^2 + x \\ \left(\forall x \,.\, \left(x^2 < y + x^3\right)\right)[x + y/y] &= \forall z \,.\, \left(z^2 < (x+y) + z^3\right).\end{aligned}$$

Notice that in the third example we first renamed the bound variable $x$ to $z$ in order to avoid a capture by $\forall$.

Substitution is simple to explain in terms of trees. Assuming Barendregt's convention, the substitution $t[u/x]$ means that in the tree $t$ we replace the leaves labeled $x$ by copies of the tree $u$. Thus a substitution never changes the structure of the tree–it only "grows" new subtrees in places where the substituted variables occur as leaves.

Substitution satisfies the distributive law

$$(t[u/x])[v/y] = (t[v/y])[u[v/y]/x],$$

provided $x$ and $y$ are distinct variables. There is also a corresponding multivariate version which is written the same way with a slight abuse of vector notation:

$$(t[\vec{u}/\vec{x}])[\vec{v}/\vec{y}] = (t[\vec{v}/\vec{y}])[\vec{u}[\vec{v}/\vec{y}]/\vec{x}].$$

## A.4 Judgments and deductive systems

A formal system, such as first-order logic or type theory, concerns itself with *judgments*. There are many kinds of judgments, such as:

- The most common judgments are equations and other logical statements. We distinguish a formula $\phi$ and the judgment "$\phi$ holds" by writing the latter as

  $$\vdash \phi\ .$$

  The symbol $\vdash$ is generally used to indicate judgments.

- Typing judgments
$$\vdash t : A$$
expressing the fact that a term $t$ has type $A$. This is not to be confused with the set-theoretic statement $t \in u$ which says that individuals $t$ and $u$ (of type "set") are in relation "element of" $\in$.

- Judgments expressing the fact that a certain entity is well formed. A typical example is a judgment
$$\vdash x_1 : A_1, \ldots, x_n : A_n \quad \texttt{ctx}$$
which states that $x_1 : A_1, \ldots, x_n : A_n$ is a well-formed context. This means that $x_1, \ldots, x_n$ are distinct variables and that $A_1, \ldots, A_n$ are well-formed types. This kind of judgement is often omitted and it is tacitly assumed that whatever entities we deal with are in fact well-formed.

A *hypothetical judgement* has the form

$$H_1, \ldots, H_n \vdash C$$

and means that hypotheses $H_1, \ldots, H_n$ entail consequence $C$ (with respect to a given decuctive system). We may also add a typing context to get a general form of judgment

$$x_1 : A_1, \ldots, x_n : A_n \mid H_1, \ldots, H_m \vdash C.$$

This should be read as: "if $x_1, \ldots, x_n$ are variables of types $A_1, \ldots, A_n$, respectively, then hypotheses $H_1, \ldots, H_m$ entail conclusion $C$." For our purposes such contexts will suffice, but you should not be surprised to see other kinds of judgments in logic.

A *deductive system* is a set of inference rules for deriving judgments. A typical inference rule has the form

$$\frac{J_1 \quad J_2 \quad \cdots \quad J_n}{J}\ C$$

This means that we can infer judgment $J$ if we have already derived judgments $J_1, \ldots, J_n$, provided that the optional side-condition $C$ is satisfied. An *axiom* is an inference rule of the form

$$\frac{}{J}$$

A *two-way rule*

$$\frac{J_1 \quad J_2 \quad \cdots \quad J_n}{\overline{K_1 \quad K_2 \quad \cdots \quad K_m}}$$

is a combination of $n + m$ inference rules stating that we may infer each $K_i$ from $J_1, \ldots, J_n$ and each $J_i$ from $K_1, \ldots, K_m$.

A *derivation* of a judgment $J$ is a finite tree whose root is $J$, the nodes are inference rules, and the leaves are axioms. An example is presented in the next subsection.

The set of all judgments that hold in a given deductive system is generated inductively by starting with the axioms and applying inference rules.

## A.5 Example: Equational reasoning

Equational reasoning is so straightforward that one almost doesn't notice it, consisting mainly, as it does, of "substituting equals for equals". The only judgements are equations between terms, $s = t$, which consist of function symbols, constants, and variables. The inference rules are just the usual ones making $s = t$ a congruence relation on the terms. More formally, we have the following specification of what may be called the *equational calculus*.

$$\begin{aligned} \text{Variable } v &::= x \mid y \mid z \mid \cdots \\ \text{Constant symbol } c &::= \mathtt{c}_1 \mid \mathtt{c}_2 \mid \cdots \\ \text{Function symbol } f^k &::= \mathtt{f}_1^{k_1} \mid \mathtt{f}_2^{k_2} \mid \cdots \\ \text{Term } t &::= v \mid c \mid f^k(t_1, \ldots, t_k) \end{aligned}$$

The superscript on the function symbol $f^k$ indicates the arity.

The equational calculus has just one form of judgement

$$x_1, \ldots, x_n \mid t_1 = t_2$$

where $x_1, \ldots, x_n$ is a *context* consisting of distinct variables, and the variables in the equation must occur among the ones listed in the context.

There are four inference rules for the equational calculus. They may be assumed to leave the contexts unchanged, which may therefore be omitted.

$$\frac{}{t = t} \qquad \frac{t_1 = t_2}{t_2 = t_1} \qquad \frac{t_1 = t_2,\ t_2 = t_3}{t_1 = t_3} \qquad \frac{t_1 = t_2,\ t_3 = t_4}{t_1[t_3/x] = t_2[t_4/x]}$$

An *equational theory* $\mathbb{T}$ consists of a set of constant and function symbols (with arities), and a set of equations, called *axioms*.

## A.6 Example: Predicate calculus

We spell out the details of single-sorted predicate calculus and first-order theories. This is the most common deductive system taught in classical courses on logic.

The predicate calculus has the following syntax:

$$\begin{aligned} \text{Variable } v &::= x \mid y \mid z \mid \cdots \\ \text{Constant symbol } c &::= \mathtt{c}_1 \mid \mathtt{c}_2 \mid \cdots \\ \text{Function symbol}^4\ f^k &::= \mathtt{f}_1^{k_1} \mid \mathtt{f}_2^{k_2} \mid \cdots \\ \text{Term } t &::= v \mid c \mid f^k(t_1, \ldots, t_k) \\ \text{Relation symbol } R^m &::= \mathtt{R}_1^{m_1} \mid \mathtt{R}_2^{m_2} \mid \cdots \\ \text{Formula } \phi &::= \bot \mid \top \mid R^m(t_1, \ldots, t_m) \mid t_1 = t_2 \mid \\ &\qquad \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \Rightarrow \phi_2 \mid \neg\phi \mid \forall\, x \,.\, \phi \mid \exists\, x \,.\, \phi. \end{aligned}$$

The variable $x$ is bound in $\forall\, x \,.\, \phi$ and $\exists\, x \,.\, \phi$.

The predicate calculus has one form of judgement

$$x_1, \dots, x_n \mid \phi_1, \dots, \phi_m \vdash \phi$$

where $x_1, \dots, x_n$ is a *context* consisting of distinct variables, $\phi_1, \dots, \phi_m$ are *hypotheses* and $\phi$ is the *conclusion*. The free variables in the hypotheses and the conclusion must occur among the ones listed in the context. We abbreviate the context with $\Gamma$ and $\Phi$ with hypotheses. Because most rules leave the context unchanged, we omit the context unless something interesting happens with it.

The following inference rules are given in the form of adjunctions. See Appendix **??** for the more usual formulation in terms of introduction an elimination rules.

$$\frac{}{\phi_1, \dots, \phi_m \vdash \phi_i} \qquad \frac{}{\Phi \vdash \top} \qquad \frac{}{\Phi, \bot \vdash \phi}$$

$$\begin{array}{c} \Phi \vdash \phi_1 \quad \Phi \vdash \phi_2 \\ \hline\hline \Phi \vdash \phi_1 \land \phi_2 \end{array} \qquad \begin{array}{c} \Phi, \phi_1 \vdash \psi \quad \Phi, \phi_2 \vdash \psi \\ \hline\hline \Phi, \phi_1 \lor \phi_2 \vdash \psi \end{array} \qquad \begin{array}{c} \Phi, \phi_1 \vdash \phi_2 \\ \hline\hline \Phi \vdash \phi_1 \Rightarrow \phi_2 \end{array}$$

$$\begin{array}{c} \Gamma, x, y \mid \Phi, x = y \vdash \phi \\ \hline\hline \Gamma, x \mid \Phi \vdash \phi[x/y] \end{array} \qquad \begin{array}{c} \Gamma, x \mid \Phi, \phi \vdash \psi \\ \hline\hline \Gamma \mid \Phi, \exists\, x \,.\, \phi \vdash \psi \end{array} \qquad \begin{array}{c} \Gamma, x \mid \Phi \vdash \phi \\ \hline\hline \Gamma \mid \Phi \vdash \forall\, x \,.\, \phi \end{array}$$

The equality rule implicitly requires that $y$ does not appear in $\Phi$, and the quantifier rules implicitly require that $x$ does not occur freely in $\Phi$ and $\psi$ because the judgments below the lines are supposed to be well formed.

Negation $\neg\phi$ is defined to be $\phi \Rightarrow \bot$. To obtain *classical* logic we also need the law of excluded middle,

$$\frac{}{\Phi \vdash \phi \lor \neg\phi}$$

Comment on the fact that contraction and weakening are admissible.

Give an example of a derivation.

A *first-order theory* $\mathbb{T}$ consists of a set of constant, function and relation symbols with corresponding arities, and a set of formulas, called *axioms*.

Give examples of a first-order theories.

# Bibliography

[ALR03] J. Adamek, F. W. Lawvere, and J. Rosiky. On the duality between varieties and algebraic theories. *Algebra Universalis*, 49:35–49, 2003.

[AR94] Jiri Adamek and Jiri Rosicky. *Locally Presentable and Accessible Categories.* Number 189 in London Mathematical Society Lecture Notes. Cambridge University Press, 1994.

[ARV10] J. Adamek, J. Rosicky, and E.M. Vitale. *Algebraic Theories.* Cambridge University Press, 2010.

[Bor94] F. Borceux. *Handbook of Categorical Algebra II. Categories and Structures*, volume 51 of *Encyclopedia of Mathematics and Its Applications.* Cambridge University Press, 1994.

[Lan71] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, New York, 1971.

[Law63a] F. W. Lawvere. Functorial semantics of algebraic theories. Ph.D. thesis, Columbia University, 1963.

[Law63b] F. W. Lawvere. Functorial semantics of algebraic theories. *Proc. Nat. Acad. Sci.*, 50:869–872, 1963.

[Law65] F. W. Lawvere. Algebraic theories, algebraic categories, and algebraic functors. In *The Theory of Models*, pages 413–418. North-Holland, 1965.

[Law69] F.W. Lawvere. Adjointness in foundations. *Dialectica*, 23:281–296, 1969.

[McC93] W. McCune. Single axioms for groups and abelian groups with various operations. *Journal of Automated Reasoning*, 10(1):1–13, 1993.

[MR91] Ieke Moerdijk and Gonzalo Reyes. *Models for Smooth Infinitesimal Analysis.* Springer-Verlag, New York, 1991.