Introduction to Categorical Logic

[DRAFT: MARCH 3, 2022]

Steve Awodey Andrej Bauer

Contents

Chapter 2

Propositional Logic

Propositional logic is the logic of propositional connectives like $p \wedge q$ and $p \Rightarrow q$. As was the case for algebraic theories, the general approach will be to determine suitable categorical structures to model the logical operations, and then use categories with such structure to represent (abstract) propositional theories. Adjoints will play a special role, as we will describe the basic logical operations as such. We again show that the semantics is "functorial", meaning that the models of a theory are functors that preserve the categorical structure. We will show that there are classifying categories for all propositional theories, as was the case for the algebraic theories that we have already met.

A more abstract, algebraic perspective will then relate the propositional case of syntax-semantics duality with classical Stone duality for Boolean algebras, and related results from lattice theory will provide an algebraic treatment of Kripke semantics for intuitionistic (and modal) propositional logic.

2.1 Propositional calculus

Before going into the details of the categorical approach, we first briefly review the propositional calculus from a conventional point of view, as we did for algebraic theories. We focus first on the *classical* propositional logic, before considering the intuitionisitc case in section **??**.

In the style of Section A.1, we have the following (abstract) syntax for (propositional) formulas:

$$\begin{aligned} \text{Propositional variable } p &::= \mathsf{p}_1 \mid \mathsf{p}_2 \mid \mathsf{p}_3 \mid \cdots \\ \text{Propositional formula } \phi &::= p \mid \top \mid \bot \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \Rightarrow \phi_2 \mid \phi_1 \Leftrightarrow \phi_2 \end{aligned}$$

An example of a formula is therefore $\big(\mathsf{p}_3 \Leftrightarrow ((((\neg\mathsf{p}_1) \vee (\mathsf{p}_2 \wedge \bot)) \vee \mathsf{p}_1) \Rightarrow \mathsf{p}_3)\big)$. We will make use of the usual conventions for parenthesis, with binding order $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$. Thus e.g. the foregoing may also be written unambiguously as $\mathsf{p}_3 \Leftrightarrow \neg\mathsf{p}_1 \vee \mathsf{p}_2 \wedge \bot \vee \mathsf{p}_1 \Rightarrow \mathsf{p}_3$.

Natural deduction

The system of *natural deduction* for propositional logic has one form of judgement

$$\mathtt{p}_1, \ldots, \mathtt{p}_n \mid \phi_1, \ldots, \phi_m \vdash \phi$$

where $\mathtt{p}_1, \ldots, \mathtt{p}_n$ is a *context* consisting of distinct propositional variables, the formulas $\phi_1, \ldots, \phi_m$ are the *hypotheses* and ϕ is the *conclusion*. The variables in the hypotheses and the conclusion must occur among those listed in the context. The hypotheses are regarded as a (finite) set; so they are unordered, have no repetitions, and may be empty. We may abbreviate the context of variables by Γ, and we often omit it.

Deductive entailment (or *derivability*) $\Phi \vdash \phi$ is thus a relation between finite sets of formulas Φ and single formulas ϕ. It is defined as the smallest such relation satisfying the following rules:

1. Hypothesis:
$$\frac{}{\Phi \vdash \phi} \ \text{ if } \phi \text{ occurs in } \Phi$$

2. Truth:
$$\frac{}{\Phi \vdash \top}$$

3. Falsehood:
$$\frac{\Phi \vdash \bot}{\Phi \vdash \phi}$$

4. Conjunction:
$$\frac{\Phi \vdash \phi \qquad \Phi \vdash \psi}{\Phi \vdash \phi \wedge \psi} \qquad \frac{\Phi \vdash \phi \wedge \psi}{\Phi \vdash \phi} \qquad \frac{\Phi \vdash \phi \wedge \psi}{\Phi \vdash \psi}$$

5. Disjunction:
$$\frac{\Phi \vdash \phi}{\Phi \vdash \phi \vee \psi} \qquad \frac{\Phi \vdash \psi}{\Phi \vdash \phi \vee \psi} \qquad \frac{\Phi \vdash \phi \vee \psi \qquad \Phi, \phi \vdash \theta \qquad \Phi, \psi \vdash \theta}{\Phi \vdash \theta}$$

6. Implication:
$$\frac{\Phi, \phi \vdash \psi}{\Phi \vdash \phi \Rightarrow \psi} \qquad \frac{\Phi \vdash \phi \Rightarrow \psi \qquad \Phi \vdash \phi}{\Phi \vdash \psi}$$

For the purpose of deduction, we define $\neg\phi := \phi \Rightarrow \bot$ and $\phi \Leftrightarrow \psi := (\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$. To obtain *classical* logic we need only include one of the following additional rules.

7. Classical logic:
$$\frac{}{\Phi \vdash \phi \vee \neg\phi} \qquad \frac{\Phi \vdash \neg\neg\phi}{\Phi \vdash \phi}$$

A *proof* of $\Phi \vdash \phi$ is a *finite* tree built from the above inference rules whose root is $\Phi \vdash \phi$. For example, here is a proof of $\phi \vee \psi \vdash \psi \vee \phi$ using the disjunction rules:

$$\dfrac{\dfrac{}{\phi \vee \psi \vdash \phi \vee \psi} \qquad \dfrac{\dfrac{}{\phi \vee \psi, \phi \vdash \phi}}{\phi \vee \psi, \phi \vdash \psi \vee \phi} \qquad \dfrac{\dfrac{}{\phi \vee \psi, \psi \vdash \psi}}{\phi \vee \psi, \psi \vdash \psi \vee \phi}}{\phi \vee \psi \vdash \psi \vee \phi}$$

A judgment $\Phi \vdash \phi$ is *provable* if there exists a proof of it. Observe that every proof has at its leaves either the rule for $\top$ or a hypothesis.

Exercise 2.1.1. Derive each of the two classical rules (2.1), called *excluded middle* and *double negation*, from the other.

2.2 Truth values

The idea of an axiomatic system of deductive, logical reasoning goes to back to Frege, who gave the first such system for propositional calculus (and more) in his *Begriffsschrift* of 1879. The question soon arose whether Frege's rules (or rather, their derivable consequences – it was clear that one could chose the primitive basis in different but equivalent ways) were correct, and if so, whether they were *all* the correct ones. An ingenious solution was proposed by Russell's student Wittgenstein, who came up with an entirely different way of singling out a set of "valid" propositional formulas in terms of assignments of truth values to the variables occurring in them. He interpreted this as showing that logical validity was really a matter of the logical structure of a proposition, and not depedent on any particular system of derivations. The same idea seems to have been had independently by Post, who proved that the valid propositional formulas coincide with the ones derivable in Whitehead and Russell's *Principia Mathematica* (which is propositionally equivalent to Frege's system), a fact that we now refer to as the *soundness* and *completeness* of propositional logic.

In more detail, let a *valuation* v be an assignment of a "truth-value" $0, 1$ to each propositional variable, $v(\mathtt{p}_n) \in \{0, 1\}$. We can then extend the valuation to all propositional formulas $[\![\phi]\!]^v$ by the recursion,

$$\begin{aligned}
[\![\mathtt{p}_n]\!]^v &= v(\mathtt{p}_n) \\
[\![\top]\!]^v &= 1 \\
[\![\bot]\!]^v &= 0 \\
[\![\neg\phi]\!]^v &= 1 - [\![\phi]\!]^v \\
[\![\phi \wedge \psi]\!]^v &= \min([\![\phi]\!]^v, [\![\psi]\!]^v) \\
[\![\phi \vee \psi]\!]^v &= \max([\![\phi]\!]^v, [\![\psi]\!]^v) \\
[\![\phi \Rightarrow \psi]\!]^v &= 1 \text{ iff } [\![\phi]\!]^v \leq [\![\psi]\!]^v \\
[\![\phi \Leftrightarrow \psi]\!]^v &= 1 \text{ iff } [\![\phi]\!]^v = [\![\psi]\!]^v
\end{aligned}$$

This is sometimes expressed using the "semantic consequence" notation $v \vDash \phi$ to mean that $[\![\phi]\!]^v = 1$. Then the above specification takes the form:

$$
\begin{array}{rll}
v \vDash \top & \text{always} & \\
v \vDash \bot & \text{never} & \\
v \vDash \neg\phi & \text{iff} & v \nvDash \phi \\
v \vDash \phi \wedge \psi & \text{iff} & v \vDash \phi \text{ and } v \vDash \psi \\
v \vDash \phi \vee \psi & \text{iff} & v \vDash \phi \text{ or } v \vDash \psi \\
v \vDash \phi \Rightarrow \psi & \text{iff} & v \vDash \phi \text{ implies } v \vDash \psi \\
v \vDash \phi \Leftrightarrow \psi & \text{iff} & v \vDash \phi \text{ iff } v \vDash \psi
\end{array}
$$

Finally, ϕ is *valid*, written $\vDash \phi$, is defined by,

$$\vDash \phi \quad \text{iff} \quad v \vDash \phi \text{ for all } v.$$

And, more generally, we define $\phi_1, ..., \phi_n$ *semantically entails* ϕ, written

$$\phi_1, ..., \phi_n \vDash \ \phi, \tag{2.1}$$

to mean that for all valuations v such that $v \vDash \phi_k$ for all k, also $v \vDash \phi$.

Given a formula in context $\Gamma \mid \phi$ and a valuation v for the variables in Γ, one can check whether $v \vDash \phi$ using a *truth table*, which is a systematic way of calculating the value of $[\![\phi]\!]^v$. For example, under the assignment $v(\mathtt{p}_1) = 1, v(\mathtt{p}_2) = 0, v(\mathtt{p}_3) = 1$ we can calculate $[\![\phi]\!]^v$ for $\phi = \big(\mathtt{p}_3 \Leftrightarrow ((((\neg\mathtt{p}_1) \vee (\mathtt{p}_2 \wedge \bot)) \vee \mathtt{p}_1) \Rightarrow \mathtt{p}_3)\big)$ as follows.

$\mathtt{p}_1$	$\mathtt{p}_2$	$\mathtt{p}_3$	$\mathtt{p}_3$	$\Leftrightarrow$	$\neg$	$\mathtt{p}_1$	$\vee$	$\mathtt{p}_2$	$\wedge$	$\bot$	$\vee$	$\mathtt{p}_1$	$\Rightarrow$	$\mathtt{p}_3$
1	0	1	1	1	0	1	0	0	0	0	1	1	1	1

The value of the formula ϕ under the valuation v is then the value in the column under the main connective, in this case $\Leftrightarrow$, and thus $[\![\phi]\!]^v = 1$.

Displaying all 2^3 valuations for the context $\Gamma = \mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_3$, therefore results in a table that checks for validity of ϕ,

$\mathtt{p}_1$	$\mathtt{p}_2$	$\mathtt{p}_3$	$\mathtt{p}_3$	$\Leftrightarrow$	$\neg$	$\mathtt{p}_1$	$\vee$	$\mathtt{p}_2$	$\wedge$	$\bot$	$\vee$	$\mathtt{p}_1$	$\Rightarrow$	$\mathtt{p}_3$
1	1	1	.	1	...									
1	1	0	.	1		...								
1	0	1	1	1	0	1	0	0	0	0	1	1	1	1
1	0	0	.	1				...						
0	1	1	.	1					...					
0	1	0	.	1						...				
0	0	1	.	1							...			
0	0	0	.	1								...		

In this case, working out the other rows shows that ϕ is indeed valid, thus $\vDash \phi$.

Theorem 2.2.1 (Soundness and Completeness of Propositional Calculus)**.** *Let Φ be any set of formulas and ψ any formula, then*

$$\Phi \vdash \psi \iff \Phi \vDash \psi.$$

In particular, for any propositional formula ϕ we have

$$\vdash \phi \iff \vDash \phi.$$

Thus derivability and validity coincide.

Proof. Let us sketch the usual proof, for later reference.

(*Soundness*:) First assume $\Phi \vdash \psi$, meaning there is a finite derivation of ψ, all of the hypotheses of which are in the set Φ. Take a valuation v such that $v \vDash \Phi$, meaning that $v \vDash \phi$ for all $\phi \in \Phi$. Observe that for each rule of inference, for any valuation v, if $v \vDash \vartheta$ for all the hypotheses of the rule, then $v \vDash \gamma$ for the conclusion. By induction on the derivations therefore $v \vDash \phi$.

(*Competeness*:) Suppose that $\Phi \nvdash \psi$, then $\Phi, \neg\psi \nvdash \bot$ (using double negation elimination). By Lemma 2.2.2 below, there is a valuation v such that $v \vDash \{\Phi, \neg\psi\}$. Thus in particular $v \vDash \Phi$ and $v \nvDash \psi$, therefore $\Phi \nvDash \psi$. □

The key lemma is this:

Lemma 2.2.2 (Model Existence)**.** *A set Φ of formulas is* consistent*, $\Phi \nvdash \bot$, just if it has a* model*, i.e. a valuation v such that $v \vDash \Phi$.*

Proof. Let Φ be any consistent set of formulas. We extend $\Phi \subseteq \Psi$ to one that is *maximally consistent*, meaning that for every formula ψ, either $\psi \in \Psi$ or $\neg\psi \in \Psi$ and not both. Enumerate the formulas $\phi_0, \phi_1, ...$, and let,

$$\begin{aligned} \Phi_0 &= \Phi, \\ \Phi_{n+1} &= \Phi_n \cup \phi_n \text{ if consistent, else } \Phi_n, \\ \Psi &= \textstyle\bigcup_n \Phi_n. \end{aligned}$$

Now for each propositional variable $\mathtt{p}$, define $v(\mathtt{p}) = 1$ just if $\mathtt{p} \in \Psi$. □

2.3 Boolean algebra

There is of course another apporach to propositional logic, which also goes back to the 19th century, namely that of Boolean algebra, which draws on the analogy between the propositional operations and the arithmetical ones.

Definition 2.3.1. A *Boolean algebra* is a set B equipped with the operations:

$$\begin{aligned} 0, 1 &: 1 \to B \\ \neg &: B \to B \\ \wedge, \vee &: B \times B \to B \end{aligned}$$

satisfying the following equations:

$$
\begin{aligned}
x \vee x &= x & x \wedge x &= x \\
x \vee y &= y \vee x & x \wedge y &= y \wedge x \\
x \vee (y \vee z) &= (x \vee y) \vee z & x \wedge (y \wedge z) &= (x \wedge y) \wedge z \\
x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) & x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \\
0 \vee x &= x & 1 \wedge x &= x \\
1 \vee x &= 1 & 0 \wedge x &= 0 \\
\neg(x \vee y) &= \neg x \wedge \neg y & \neg(x \wedge y) &= \neg x \vee \neg y \\
x \vee \neg x &= 1 & x \wedge \neg x &= 0
\end{aligned}
$$

This is of course an algebraic theory, like those considered in the previous chapter. Familiar examples of Boolean algebras are $2 = \{0, 1\}$, with the usual operations, and more generally, any powerset $\mathcal{P}X$, with the set-theoretic operations $A \vee B = A \cup B$, etc. (indeed, $2 = \mathcal{P}1$ is a special case.).

Exercise 2.3.2. Show that the free Boolean algebra $B(n)$ on n-many generators is the double powerset $\mathcal{P}\mathcal{P}(n)$, and determine the free functor on finite sets.

One can use equational reasoning in Boolean algebra as an alternative to the deductive propositional calculus as follows. For a propositional formula in context $\Gamma \mid \phi$, let us say that ϕ is *equationally provable* if we can prove $\phi = 1$ by equational reasoning (Section A.5), from the laws of Boolean algebras above. More generally, for a set of formulas Φ and a formula ψ let us define the *ad hoc* relation of *equational provability*,

$$\Phi \vdash^{=} \psi \tag{2.2}$$

to mean that $\psi = 1$ can be proven equationally from (the Boolean equations and) the set of all equations $\phi = 1$, for $\phi \in \Phi$. Since we don't have any laws for the connectives $\Rightarrow$ or $\Leftrightarrow$, let us replace them with their Boolean equivalents, by adding the equations:

$$
\begin{aligned}
&\phi \Rightarrow \psi = \neg\phi \vee \psi\,, \\
&\phi \Leftrightarrow \psi = (\neg\phi \vee \psi) \wedge (\neg\psi \vee \phi)\,.
\end{aligned}
$$

For example, here is an equational proof of $(\phi \Rightarrow \psi) \vee (\psi \Rightarrow \phi)$.

$$
\begin{aligned}
(\phi \Rightarrow \psi) \vee (\psi \Rightarrow \phi) &= (\neg\phi \vee \psi) \vee (\neg\psi \vee \phi) \\
&= \neg\phi \vee (\psi \vee (\neg\psi \vee \phi)) \\
&= \neg\phi \vee ((\psi \vee \neg\psi) \vee \phi) \\
&= \neg\phi \vee (1 \vee \phi) \\
&= \neg\phi \vee 1 \\
&= 1 \vee \neg\phi \\
&= 1
\end{aligned}
$$

Thus,

$$\vdash^{=} (\phi \Rightarrow \psi) \vee (\psi \Rightarrow \phi)\,.$$

We now ask: *What is the relationship between equational provability $\Phi \vdash^{=} \phi$, deductive entailment $\Phi \vdash \phi$, and semantic entailment $\Phi \vDash \phi$?*

Exercise 2.3.3. Using equational reasoning, show that every propositional formula ϕ has both a *conjunctive* $\phi^{\wedge}$ and a *disjunctive* $\phi^{\vee}$ *Boolean normal form* such that:

1. The formula $\phi^{\vee}$ is an n-fold disjunction of m-fold conjunctions of *positive* $\mathtt{p_i}$ or *negative* $\neg\mathtt{p_j}$ propositional variables,

   $$\phi^{\vee} \;=\; (\mathtt{q}_{11} \wedge ... \wedge \mathtt{q}_{1m_1}) \vee ... \vee (\mathtt{q}_{n1} \wedge ... \wedge \mathtt{q}_{nm_n})\,, \qquad \mathtt{q}_{ij} \in \{\mathtt{p}_{ij}, \neg\mathtt{p}_{ij}\}\,,$$

   and $\phi^{\wedge}$ is the same, but with the roles of $\vee$ and $\wedge$ reversed.

2. Both

   $$\vdash^{=} \phi \Leftrightarrow \phi^{\vee} \qquad \text{and} \qquad \vdash^{=} \phi \Leftrightarrow \phi^{\wedge}\,.$$

Exercise 2.3.4. Using Exercise 2.3.3, show that for every propositional formula ϕ, equational provability is equivalent to semantic validity,

$$\vdash^{=} \phi \iff \vDash \phi\,.$$

Hint: Put ϕ into conjunctive normal form and read off a truth valuation that falsifies it, if there is one.

Exercise 2.3.5. A Boolean algebra can be partially ordered by defining $x \leq y$ as

$$x \leq y \iff x \vee y = y \qquad \text{or equivalently} \qquad x \leq y \iff x \wedge y = x\,.$$

Thus a Boolean algebra is a (poset) category. Show that as a category, a Boolean algebra has all finite limits and colimits and is cartesian closed, and that a finitely complete and cocomplete cartesian closed poset is a Boolean algebra just if it satisfies $x = (x \Rightarrow 0) \Rightarrow 0$, where, as before, we define $x \Rightarrow y := \neg x \vee y$. Finally, show that homomorphisms of Boolean algebras $f : B \to B'$ are the same thing as functors (i.e. monotone maps) that preserve all finite limits and colimits.

2.4 Lawvere duality for Boolean algebras

Let us apply the machinery of algebraic theories from Chapter **??** to the algebraic theory of Boolean algebras and see what we get. The algebraic theory $\mathbb{B}$ of Boolean algebras is a finite product (FP) category with objects $1, B, B^2, ...$, containing a Boolean algebra $\mathcal{B}$, with underlying object $|\mathcal{B}| = B$. By Theorem **??**, $\mathbb{B}$ has the universal property that finite product preserving (FP) functors from $\mathbb{B}$ into any FP-category $\mathbb{C}$ correspond (pseudo-)naturally to Boolean algebras in $\mathbb{C}$,

$$\mathsf{Hom}_{\mathsf{FP}}(\mathbb{B},\mathbb{C}) \simeq \mathsf{BA}(\mathbb{C}). \tag{2.3}$$

The correspondence is mediated by evaluating an FP functor $F : \mathbb{B} \to \mathbb{C}$ at (the underlying structure of) the Boolean algebra $\mathcal{B}$ to get a Boolean algebra $F(\mathcal{B}) = \mathsf{BA}(F)(\mathcal{B})$ in $\mathbb{C}$:

$$\frac{F : \mathbb{B} \longrightarrow \mathbb{C} \qquad \mathsf{FP}}{\overline{\;F(\mathcal{B}) \qquad\quad \mathsf{BA}(\mathbb{C})\;}}$$

We call $\mathcal{B}$ the *universal Boolean algebra.* Given a Boolean algebra $\mathcal{A}$ in $\mathbb{C}$, we write

$$\mathcal{A}^{\sharp} : \mathbb{B} \longrightarrow \mathbb{C}$$

for the associated *classifying functor*. By the equivalence of categories (2.3), we have isos,

$$\mathcal{A}^{\sharp}(\mathcal{B}) \cong \mathcal{A}, \qquad F(\mathcal{B})^{\sharp} \cong F.$$

And in particular, $\mathcal{B}^{\sharp} \cong 1_{\mathbb{B}} : \mathbb{B} \to \mathbb{B}$.

By Lawvere duality, Corollary **??**, we know that $\mathbb{B}^{\mathsf{op}}$ can be identified with a full subcategory $\mathsf{mod}(\mathbb{B})$ of $\mathbb{B}$-models in Set (i.e. Boolean algebras),

$$\mathbb{B}^{\mathsf{op}} = \mathsf{mod}(\mathbb{B}) \hookrightarrow \mathsf{Mod}(\mathbb{B}) = \mathsf{BA}(\mathsf{Set}), \tag{2.4}$$

namely, that consisting of the finitely generated free Boolean algebras $F(n)$. Composing (2.4) and (2.3), we have an embedding of $\mathbb{B}^{\mathsf{op}}$ into the functor category,

$$\mathbb{B}^{\mathsf{op}} \hookrightarrow \mathsf{BA}(\mathsf{Set}) \simeq \mathsf{Hom}_{\mathsf{FP}}(\mathbb{B},\mathsf{Set}) \hookrightarrow \mathsf{Set}^{\mathbb{B}}, \tag{2.5}$$

which, up to isomorphism, is just the (contravariant) Yoneda embedding, taking $B^n \in \mathbb{B}$ to the covariant representable functor $\mathsf{y}^{\mathbb{B}}(B^n) = \mathsf{Hom}_{\mathbb{B}}(B^n, -)$ (cf. Theorem **??**).

Now consider provability of equations between terms $\phi : B^k \to B$ in the theory $\mathbb{B}$, which are essentially the same as propositional formulas in context $(\mathsf{p}_1, ..., \mathsf{p}_k \mid \phi)$ modulo $\mathbb{B}$-provable equality. The universal Boolean algebra $\mathcal{B}$ is logically generic, in the sense that for any such formulas ϕ, ψ, we have $\mathcal{B} \vDash \phi = \psi$ just if $\mathbb{B} \vdash \phi = \psi$ (Proposition **??**). The latter condition is equational provability from the axioms for Boolean algebras, which is just what was used in the definition of $\vdash^{=} \phi$ (cf. 2.2). Thus, in particular,

$$\vdash^{=} \phi \iff \mathbb{B} \vdash \phi = 1 \iff \mathcal{B} \vDash \phi = 1.$$

As we showed in Proposition **??**, the image of the universal model $\mathcal{B}$ under the (FP) *covariant* Yoneda embedding,

$$\mathsf{y}_{\mathbb{B}} : \mathbb{B} \to \mathsf{Set}^{\mathbb{B}^{\mathsf{op}}}$$

is also a logically generic model, with underlying object $|\mathsf{y}_{\mathbb{B}}(\mathcal{B})| = \mathsf{Hom}_{\mathbb{B}}(-, B)$. By Proposition **??** we can use that fact to restrict attention to Boolean algebras in Set, and in particular, to the finitely generated free ones $F(n)$, when testing for equational provability. Specifically, using the (FP) evaluation functors $\mathsf{eval}_{B^n} : \mathsf{Set}^{\mathbb{B}^{\mathsf{op}}} \to \mathsf{Set}$ for all objects $B^n \in \mathbb{B}$, we can extend the above reasoning as follows:

$$\begin{aligned} \vdash^{=} \phi &\iff \mathbb{B} \vdash \phi = 1 \\ &\iff \mathcal{B} \vDash \phi = 1 \\ &\iff \mathsf{y}_{\mathbb{B}}(\mathcal{B}) \vDash \phi = 1 \\ &\iff \mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(\mathcal{B}) \vDash \phi = 1 \quad \text{for all } B^n \in \mathbb{B} \\ &\iff F(n) \vDash \phi = 1 \quad \text{for all } n. \end{aligned}$$

The last step holds because the image of $\mathsf{y}_{\mathbb{B}}(\mathcal{B})$ under eval_{B^n} is the free Boolean algebra $F(n)$ (cf. Exercise **??**). Indeed, for the underlying objects we have

$$\begin{aligned} |\mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(\mathcal{B})| &\cong \mathsf{eval}_{B^n}|\mathsf{y}_{\mathbb{B}}(\mathcal{B})| \cong \mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(|\mathcal{B}|) \cong \mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(B) \cong \mathsf{y}_{\mathbb{B}}(B)(B^n) \\ &\cong \mathsf{Hom}_{\mathbb{B}}(B^n, B) \cong \mathsf{Hom}_{\mathsf{BA}^{\mathsf{op}}}(F(n), F(1)) \cong \mathsf{Hom}_{\mathsf{BA}}(F(1), F(n)) \cong |F(n)|\,. \end{aligned}$$

Thus to test for equational provability it suffices to check the equations in the free algebras $F(n)$ (which makes sense, since these are usually *defined* in terms of equational provability). We have therefore shown:

Lemma 2.4.1. *A formula in context* $\mathtt{p}_1, ..., \mathtt{p}_k \mid \phi$ *is equationally provable* $\vdash_{=} \phi$ *just in case, for every free Boolean algebra* $F(n)$*, we have* $F(n) \vDash \phi = 1$*.*

The condition $F(n) \vDash \phi = 1$ means that the equation $\phi = 1$ holds *generally* in $F(n)$, i.e. for any elements $f_1, ..., f_k \in F(n)$, we have $\phi[f_1/\mathtt{p}_1, ..., f_k/\mathtt{p}_k] = 1$, where the expression $\phi[f_1/\mathtt{p}_1, ..., f_k/\mathtt{p}_k]$ denotes the element of $F(n)$ resulting from interpreting the propositional variables $\mathtt{p}_i$ as the elements f_i and evaluating the resulting expression using the Boolean operations of $F(n)$. But now observe that the recipe:

> for any elements $f_1, ..., f_k \in F(n)$, let the expression
>
> $$\phi[f_1/\mathtt{p}_1, ..., f_k/\mathtt{p}_k] \tag{2.6}$$
>
> denote the element of $F(n)$ resulting from interpreting the propositional variables $\mathtt{p}_i$ as the elements f_i and evaluating the resulting expression using the Boolean operations of $F(n)$

describes the unique Boolean homomorphism

$$F(1) \xrightarrow{\overline{\phi}} F(k) \xrightarrow{\overline{(f_1, ..., f_k)}} F(n)\,,$$

where $\overline{(f_1, ..., f_k)} : F(k) \to F(n)$ is determined by the elements $f_1, ..., f_k \in F(n)$, and $\overline{\phi} : F(1) \to F(k)$ by the corresponding element $(\mathtt{p}_1, ..., \mathtt{p}_k \mid \phi) \in F(k)$. It is therefore equivalent to check the case $k = n$ and $f_i = \mathtt{p}_i$, i.e. the "universal case"

$$(\mathtt{p}_1, ..., \mathtt{p}_k \mid \phi) = 1 \quad \text{in } F(k)\,. \tag{2.7}$$

Finally, we then have:

Proposition 2.4.2 (Completeness of the equational propositional calculus)**.** *Equational propositional calculus is sound and complete with respect to* boolean-valued models *in* Set*, in the sense that a propositional formula ϕ is equationally provable from the laws of Boolean algebra,*

$$\vdash^{=} \phi\,,$$

just if it holds generally in any Boolean algebra (in Set*).*

Proof. By "holding generally" is meant the universal quantification of the equation over elements of a given Boolean algebra B, which is of course equivalent to saying that it holds for all elements of B, in the sense stated after the Lemma. But, as above, this is equivalent to the condition that for all $b_1, ..., b_k \in B$, for $\overline{(b_1, ..., b_k)} : F(k) \to B$ we have $\overline{(b_1, ..., b_k)}(\phi) = 1$ in B, which in turn is clearly equivalent to the previously determined "universal" condition (2.7) that $\phi = 1$ in $F(k)$. □

The analogous statement for equational entailment $\Phi \vdash_{=} \phi$ is left as an exercise.

Corollary 2.4.2 is a (very) special case of the Gödel completeness theorem for first-order logic, for *just* the equational fragment of *just* the specific theory of Boolean algebras (although, an analogous result of course holds for any other algebraic theory, and many other systems of logic can be reduced to the algebraic case). Nonetheless, it suggests another approach to the semantics of propositional logic based upon the idea of a *Boolean valuation*, generalizing the traditional truth-value semantics from Section 2.2. We pursue this idea systematically in the following section.

Exercise 2.4.3. For a formula in context $\mathtt{p}_1, ..., \mathtt{p}_k \mid \vartheta$ and a Boolean algebra $\mathcal{A}$, let the expression $\vartheta[a_1/\mathtt{p}_1, ..., a_k/\mathtt{p}_k]$ denote the element of $\mathcal{A}$ resulting from interpreting the propositional variables $\mathtt{p}_i$ in the context as the elements a_i of $\mathcal{A}$, and evaluating the resulting expression using the Boolean operations of $\mathcal{A}$. For any *finite* set of propositional formulas Φ and any formula ψ, let $\Gamma = \mathtt{p}_1, ..., \mathtt{p}_k$ be a context for (the formulas in) $\Phi \cup \{\psi\}$. Finally, recall that $\Phi \vdash^{=} \psi$ means that $\psi = 1$ is equationally provable from the set of equations $\{\phi = 1 \mid \phi \in \Phi\}$. Show that $\Phi \vdash^{=} \psi$ just if for all finitely generated free Boolean algebras $F(n)$, the following condition holds:

> For any elements $f_1, ..., f_k \in F(n)$, if $\phi[f_1/\mathtt{p}_1, ..., f_k/\mathtt{p}_k] = 1$ for all $\phi \in \Phi$, then $\psi[f_1/\mathtt{p}_1, ..., f_k/\mathtt{p}_k] = 1$.

Is it sufficient to just take $F(k)$ and its generators $\mathtt{p}_1, ..., \mathtt{p}_k$ as the $f_1, ..., f_k$? Is it equivalent to take all Boolean algebras B, rather than the finitely generated free ones $F(n)$? Determine a condition that is equivalent to $\Phi \vdash^{=} \psi$ for not necessarily finite sets Φ.

2.5 Functorial semantics for propositional logic

Considering the algebraic theory of Boolean algebras suggests the idea of a Boolean valuation of propositional logic, generalizing the truth valuations of section 2.2. This can be seen as applying the framework of functorial semantics to a different system of logic than that of finite product categories, namely that represented categorically by *poset* categories with finite products $\wedge$ *and coproducts* $\vee$ (each of these specializations could, of course, also be considered separately, giving $\wedge$-semi-lattices and categories with finite products $\times$ and coproducts $+$, respectively). Thus we are moving from the top right corner to the bottom center position in the following Hasse diagram of structured categories:

$$
\begin{tikzcd}
+ \arrow[dr, dash] \arrow[dd, dash] & & \times \arrow[dl, dash] \arrow[dd, dash] \\
& +\,,\times \arrow[dd, dash] & \\
\vee \arrow[dr, dash] & & \wedge \arrow[dl, dash] \\
& \vee\,,\wedge &
\end{tikzcd}
$$

In Chapter **??** we shall see how first-order logic results categorically from these two cases by "indexing the lower one over the upper one".

Definition 2.5.1. A *propositional theory* $\mathbb{T}$ consists of a set $V_{\mathbb{T}}$ of propositional variables, called the *basic* or *atomic propositions*, and a set $A_{\mathbb{T}}$ of propositional formulas (over $V_{\mathbb{T}}$), called the *axioms*. The *consequences* $\Phi \vdash_{\mathbb{T}} \phi$ are those judgements that are derivable by natural deduction (as in Section 2.1), from the axioms $A_{\mathbb{T}}$.

Definition 2.5.2. Let $\mathbb{T} = (V_{\mathbb{T}}, A_{\mathbb{T}})$ be a propositional theory and $\mathcal{B}$ a Boolean algebra. A *model* of $\mathbb{T}$ in $\mathcal{B}$, also called a *Boolean valuation of* $\mathbb{T}$ is an *interpretation function* $v : V_{\mathbb{T}} \to |\mathcal{B}|$ such that, for every $\alpha \in A_{\mathbb{T}}$, we have $[\![\alpha]\!]^v = 1_{\mathcal{B}}$ in $\mathcal{B}$, where the extension $[\![-]\!]^v$ of v from $V_{\mathbb{T}}$ to all formulas (over $V_{\mathbb{T}}$) is defined in the expected way, namely:

$$
\begin{aligned}
[\![\mathtt{p}]\!]^v &= v(\mathtt{p}), \quad \mathtt{p} \in V_{\mathbb{T}} \\
[\![\top]\!]^v &= 1_{\mathcal{B}} \\
[\![\bot]\!]^v &= 0_{\mathcal{B}} \\
[\![\neg\phi]\!]^v &= \neg_{\mathcal{B}}[\![\phi]\!]^v \\
[\![\phi \wedge \psi]\!]^v &= [\![\phi]\!]^v \wedge_{\mathcal{B}} [\![\psi]\!]^v \\
[\![\phi \vee \psi]\!]^v &= [\![\phi]\!]^v \vee_{\mathcal{B}} [\![\psi]\!]^v \\
[\![\phi \Rightarrow \psi]\!]^v &= \neg_{\mathcal{B}}[\![\phi]\!]^v \vee_{\mathcal{B}} [\![\psi]\!]^v
\end{aligned}
$$

Finally, let $\mathsf{Mod}(\mathbb{T}, \mathcal{B})$ be the set of all $\mathbb{T}$-models in $\mathcal{B}$. Given a Boolean homomorphism $f : \mathcal{B} \to \mathcal{B}'$, there is an induced mapping $\mathsf{Mod}(\mathbb{T}, f) : \mathsf{Mod}(\mathbb{T}, \mathcal{B}) \to \mathsf{Mod}(\mathbb{T}, \mathcal{B}')$, determined by setting $\mathsf{Mod}(\mathbb{T}, f)(v) = f \circ v$, which is clearly functorial.

Theorem 2.5.3. *The functor* $\mathsf{Mod}(\mathbb{T}) : \mathsf{BA} \to \mathsf{Set}$ *is representable, with representing Boolean algebra* $\mathcal{B}_{\mathbb{T}}$*, called the* Lindenbaum-Tarski algebra *of* $\mathbb{T}$.

Proof. We construct $\mathcal{B}_{\mathbb{T}}$ in two steps:

Step 1: Suppose first that $A_{\mathbb{T}}$ is empty, so $\mathbb{T}$ is just a set V of propositional variables. Define the *Lindenbaum-Tarski algebra* $\mathcal{B}[V]$ by

$$\mathcal{B}[V] \;=\; \{\phi \mid \phi \text{ is a formula in context } V\}/\sim$$

where the equivalence relation $\sim$ is *(deductively) provable bi-implication*,

$$\phi \sim \psi \iff \vdash \psi \Leftrightarrow \psi.$$

The operations are (well-)defined on equivalence classes by setting,

$$[\phi] \wedge [\psi] \;=\; [\phi \wedge \psi],$$

and so on. (The reader who has not seen this construction before should fill in the details!)

Step 2: In the general case $\mathbb{T} = (V_{\mathbb{T}}, A_{\mathbb{T}})$, let

$$\mathcal{B}_{\mathbb{T}} = \mathcal{B}[V_{\mathbb{T}}]/\sim_{\mathbb{T}}\,,$$

where the equivalence relation $\sim_{\mathbb{T}}$ is now $A_{\mathbb{T}}$*-provable bi-implication*,

$$\phi \sim_{\mathbb{T}} \psi \iff A_{\mathbb{T}} \vdash \psi \Leftrightarrow \psi.$$

The operations are defined as before, but now on equivalence classes $[\phi]$ modulo $A_{\mathbb{T}}$.

Now observe that the construction of $\mathcal{B}_{\mathbb{T}}$ is a variation on that of the *syntactic category* $\mathcal{C}_{\mathbb{T}}$ of the algebraic theory $\mathbb{T}$ in the sense of the previous chapter, and the statement of the theorem is its universal property as the classifying category of $\mathbb{T}$-models, namely

$$\mathsf{Mod}(\mathbb{T}, \mathcal{B}) \cong \mathsf{Hom}_{\mathsf{BA}}(\mathcal{B}_{\mathbb{T}}, \mathcal{B})\,, \tag{2.8}$$

naturally in $\mathcal{B}$. (Indeed, since $\mathsf{Mod}(\mathbb{T}, \mathcal{B})$ is now a *set* rather than a category, we can classify it up to *isomorphism* rather than equivalence of categories.) The proof of this fact is a variation on the proof of the corresponding theorem **??** from Chapter 1. Further details are given in the following Remark 2.5.4 for the interested reader. □

Remark 2.5.4 (Adjoint Rules for Propositional Calculus)**.** For the construction of the Lindenbaum-Tarski algebra $\mathcal{B}_{\mathbb{T}}$, it is convenient to reformulate the rules of inference for the propositional calculus in the following equivalent *adjoint form*:

Contexts Γ may be omitted, since the rules leave them unchanged (there is no variable binding). We may also omit hypotheses that remain unchanged. Thus e.g. the *hypothesis* rule may be written in any of the following equivalent ways.

$$\frac{}{\Gamma \mid \phi_1, \ldots, \phi_m \vdash \phi_i} \qquad \frac{}{\phi_1, \ldots, \phi_m \vdash \phi_i} \qquad \frac{}{\phi \vdash \phi}$$

The structural rules can then be stated as follows:

$$\frac{}{\phi \vdash \phi} \qquad \frac{\phi \vdash \psi \qquad \psi \vdash \vartheta}{\phi \vdash \vartheta}$$

$$\frac{\phi \vdash \vartheta}{\psi, \phi \vdash \vartheta} \qquad \frac{\phi, \phi \vdash \vartheta}{\phi \vdash \vartheta} \qquad \frac{\phi, \psi \vdash \vartheta}{\psi, \phi \vdash \vartheta}$$

The rules for the propositional connectives can be given in the following adjoint form, where the double line indicates a two-way rule (with the obvious two instances when there are two conclusions).

$$\frac{}{\phi \vdash \top} \qquad \frac{}{\bot \vdash \phi}$$

$$\frac{\vartheta \vdash \phi \quad \vartheta \vdash \psi}{\vartheta \vdash \phi \wedge \psi} \qquad \frac{\phi \vdash \vartheta \quad \psi \vdash \vartheta}{\phi \vee \psi \vdash \vartheta} \qquad \frac{\vartheta, \phi \vdash \psi}{\vartheta \vdash \phi \Rightarrow \psi}$$

For the purpose of deduction, negation $\neg\phi$ is again treated as defined by $\phi \Rightarrow \bot$ and bi-implication $\phi \Leftrightarrow \psi$ by $(\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$. For *classical* logic we also include the rule of *double negation*:

$$\frac{}{\neg\neg\phi \vdash \phi} \tag{2.9}$$

It is now obvious that the set of formulas is preordered by $\phi \vdash \psi$, and that the poset reflection agrees with the deducibility equivalence relation,

$$\phi \dashv\vdash \psi \iff \phi \sim \psi\,.$$

Moreover, $\mathcal{B}_{\mathbb{T}}$ clearly has all finite limits $\top, \wedge$ and colimits $\bot, \vee$, is cartesian closed $\wedge \dashv \Rightarrow$, and is therefore a *Heyting algebra* (see Section **??** below). The rule of double negation then makes it a Boolean algebra.

The proof of the universal property of $\mathcal{B}_{\mathbb{T}}$ is essentially the same as that for $\mathcal{C}_{\mathbb{T}}$.

Exercise 2.5.5. Fill in the details of the proof that $\mathcal{B}_{\mathbb{T}}$ is a well-defined Boolean algebra, with the universal property stated in (2.8).

Just as for the case of algebraic theories and FP categories, we now have the following corollary of the classifying theorem 2.5.3. (Note that the recipe at (2.6) for a Boolean valuation in $F(n)$ of the formula in context $\mathtt{p}_1, ..., \mathtt{p}_k \mid \phi$ is exactly a *model* in $F(n)$ of the theory $\mathbb{T} = \{\mathtt{p}_1, ..., \mathtt{p}_k\}$.)

Corollary 2.5.6. *For any set of formulas Φ and formula ϕ, derivability $\Phi \vdash \phi$ is equivalent to validity under all Boolean valuations. Therefore by Proposition 2.4.2 (and Exercise 2.4.3), we also have*

$$\Phi \vdash \phi \iff \Phi \vdash^{=} \phi\,.$$

Remark 2.5.7. If $A_{\mathbb{T}}$ is non-empty, but finite, then let

$$\alpha_{\mathbb{T}} := \bigwedge_{\alpha \in A_{\mathbb{T}}} \alpha \,.$$

We then have

$$\mathcal{B}_{\mathbb{T}} = \mathcal{B}[V_{\mathbb{T}}]/\alpha_{\mathbb{T}} \,,$$

where as usual $\mathcal{B}/b$ denotes the slice category of the Boolean algebra $\mathcal{B}$ over an element $b \in \mathcal{B}$.

Remark 2.5.8. Our definition of the Lindenbaum-Tarski algebra is given in terms of *provability*, rather than the more familiar semantic definition using (truth) valuations. The two are, of course, equivalent in light of Theorem 2.2.1, but since we intend to prove that theorem, this definition will be more useful, as it parallels that of the syntactic category $\mathcal{C}_{\mathbb{T}}$ of an algebraic theory.

Inspecting the universal property (2.8) of $\mathcal{B}_{\mathbb{T}}$ for the case $\mathcal{B}[V]$ where there are no axioms, we now have the following.

Corollary 2.5.9. *The Lindenbaum-Tarski algebra $\mathcal{B}[V]$ is the free Boolean algebra on the set V. In particular, $\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]$ is the finitely generated, free Boolean algebra $F(n)$.*

The isomorphism $\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n] \cong F(n)$ expresses the fact recorded in Corollary 2.5.6 that the relations of derivability by natural deduction $\Phi \vdash \phi$ and equational provability $\Phi \vdash^{=} \phi$ agree — answering part of the question at the end of Section **??**.

Exercise 2.5.10. Show that the Boolean algebras $\mathcal{B}_{\mathbb{T}}$ for *finite sets* $V_{\mathbb{T}}$ of variables and $A_{\mathbb{T}}$ of formulas are exactly the *finitely presented* ones.

Finally, we can use the following to finish the comparison of $\vdash \phi$ and $\vDash \phi$.

Lemma 2.5.11. *Let $\mathcal{B}$ be a* finitely presented *Boolean algebra in which $0 \neq 1$. Then there is a Boolean homomorphism*

$$h : \mathcal{B} \to \mathbb{2} \,.$$

Proof. By Exercise 2.5.10, we can assume that $\mathcal{B} = \mathcal{B}[\mathtt{p}_1...\mathtt{p}_n]/\alpha$ classifying the theory $\mathbb{T} = (\mathtt{p}_1...\mathtt{p}_n, \alpha)$. By the assumption that $0 \neq 1$ in $\mathbb{B} = \mathcal{B}[\mathtt{p}_1...\mathtt{p}_n]/\alpha$, we have $\alpha \neq 0$ in the free Boolean algebra $F(n) \cong \mathcal{B}[\mathtt{p}_1...\mathtt{p}_n]$, whence $\alpha \nvdash \bot$. Since $F(n) \cong \mathcal{P}\mathcal{P}(n)$, there is a valuation $\vartheta : \{\mathtt{p}_1...\mathtt{p}_n\} \to \mathbb{2}$ such that $[\![\alpha]\!]^{\vartheta} = 1$. This is exactly a Boolean homomorphism $\mathcal{B}[\mathtt{p}_1...\mathtt{p}_n]/\alpha \to \mathbb{2}$, as required. $\square$

Corollary 2.5.12. *For any set of formulas Φ and formula ϕ, derivability $\Phi \vdash \phi$ is equivalent to semantic entailment,*

$$\Phi \vDash \phi \iff \Phi \vdash \phi \,.$$

Proof. By 2.5.6, it suffices to show that $\Phi \vDash \phi$ is equivalent to $\Phi \vdash^{=} \phi$, but the latter we know to be equivalent to holding in all Boolean valuations in free Boolean algebras $F(n)$, and the former to holding in all *truth* valuations, i.e. Boolean valuations in $\mathbb{2}$. Thus it will suffice to embed $F(n)$ as a Boolean algebra into a powerset $\mathcal{P}X = \mathbb{2}^X$, for a set X. By Lemma 2.5.11 we can take $X = 2^n$. □

Lemma 2.5.11 will also play a key role in the Stone duality in the next section.

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.

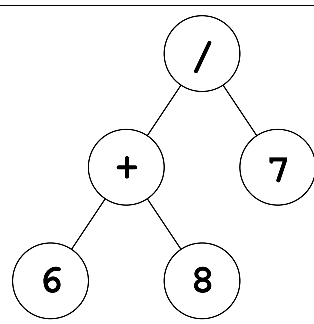

Figure A.1: The tree represented by $(6 + 8)/7$

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:

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

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 ϕ_1, ϕ_2, and ϕ 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

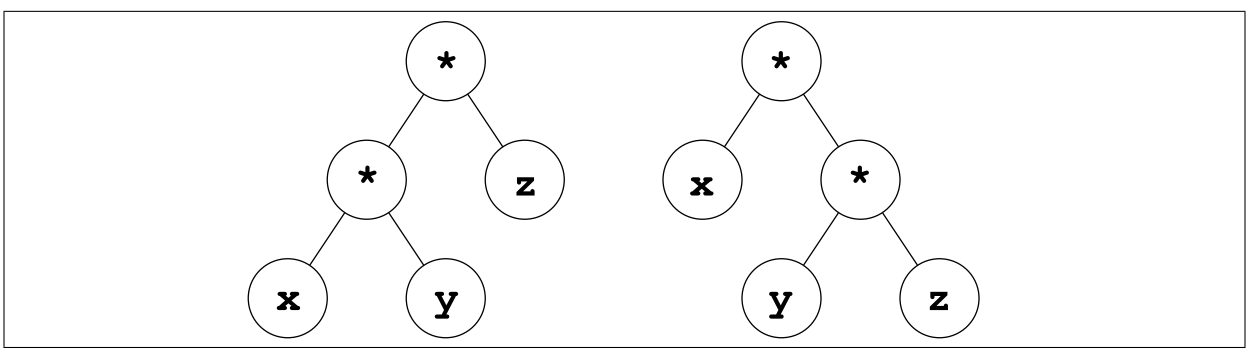

Figure A.2: Left and right associativity of $x \star y \star z$

associative, but is not associative in the usual mathematical sense.

[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. **??**,
- λ-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 *α-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 α-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:

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

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 ϕ and the judgment "ϕ 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

  $$\vdash x_1 : A_1, \ldots, x_n : A_n \quad \mathtt{ctx}$$

  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*

$$\begin{array}{cccc} J_1 & J_2 & \cdots & J_n \\ \hline\hline K_1 & K_2 & \cdots & K_m \end{array}$$

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

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

where $x_1, \ldots, x_n$ is a *context* consisting of distinct variables, $\phi_1, \ldots, \phi_m$ are *hypotheses* and ϕ 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 Γ and Φ 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.

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

$$\frac{\Phi \vdash \phi_1 \quad \Phi \vdash \phi_2}{\Phi \vdash \phi_1 \land \phi_2} \qquad \frac{\Phi, \phi_1 \vdash \psi \quad \Phi, \phi_2 \vdash \psi}{\Phi, \phi_1 \lor \phi_2 \vdash \psi} \qquad \frac{\Phi, \phi_1 \vdash \phi_2}{\Phi \vdash \phi_1 \Rightarrow \phi_2}$$

$$\frac{\Gamma, x, y \mid \Phi, x = y \vdash \phi}{\Gamma, x \mid \Phi \vdash \phi[x/y]} \qquad \frac{\Gamma, x \mid \Phi, \phi \vdash \psi}{\Gamma \mid \Phi, \exists x . \phi \vdash \psi} \qquad \frac{\Gamma, x \mid \Phi \vdash \phi}{\Gamma \mid \Phi \vdash \forall x . \phi}$$

The equality rule implicitly requires that y does not appear in Φ, and the quantifier rules implicitly require that x does not occur freely in Φ and ψ 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.